Sass, Susanne; Mitsos, Alexander; Nikolov, Nikolay I.; Tsoukalas, Angelos

Out-of-sample estimation for a branch-and-bound algorithm with growing datasets

Journal of Global Optimization

Provided in Cooperation with:
Springer Nature

Suggested Citation: Sass, Susanne; Mitsos, Alexander; Nikolov, Nikolay I.; Tsoukalas, Angelos (2025) : Out-of-sample estimation for a branch-and-bound algorithm with growing datasets, Journal of Global Optimization, ISSN 1573-2916, Springer US, New York, NY, Vol. 92, Iss. 3, pp. 615-642, https://doi.org/10.1007/s10898-025-01514-4

Journal of Global Optimization (2025) 92:615–642
https://doi.org/10.1007/s10898-025-01514-4

Out-of-sample estimation for a branch-and-bound algorithm with growing datasets

Susanne Sass[1] · **Alexander Mitsos**[1,2,3] · **Nikolay I. Nikolov**[4,5] · **Angelos Tsoukalas**[6]

Received: 10 May 2024 / Accepted: 12 June 2025 / Published online: 23 June 2025

Abstract
In [Sass et al., Eur. J. Oper. Res., 316 (1): 36 – 45, 2024], we proposed a branch-and-bound (B&B) algorithm with growing datasets for the deterministic global optimization of parameter estimation problems based on large datasets. Therein, we start the B&B algorithm with a reduced dataset and augment it until reaching the full dataset upon convergence. However, convergence may be slowed down by a gap between the lower bounds of the reduced and the original problem, in particular for noisy measurement data. Thus, we propose the use of out-of-sample estimation for improving the lower bounds calculated with reduced datasets. Based on this, we extend the deterministic approach and propose two heuristic approaches. The computational performance of all approaches is compared with the standard B&B algorithm as a benchmark based on real-world estimation problems from process systems engineering, biochemistry, and machine learning covering datasets with and without measurement noise. Our results indicate that the heuristic approaches can improve the final lower bounds on the optimal objective value without cutting off the global solution. Aside from this, we prove that resampling can decrease the variance of the lower bounds calculated based on random initial datasets. In our case study, resampling hardly affects the performance of the approaches which indicates that the B&B algorithm with growing datasets does not suffer from large variances.

Keywords Nonlinear programming · Spatial branch and bound · Parameter estimation · Overfitting · Resampling

✉ Angelos Tsoukalas
tsoukalas@rsm.nl

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

2 JARA-CSD, 52056 Aachen, Germany

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

4 Institute of Statistics, RWTH Aachen University, 52056 Aachen, Germany

5 Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria

6 Department of Technology and Operations Management, RSM Erasmus University Rotterdam, 3062 PA Rotterdam, Netherlands

1 Introduction

The validity of parameter estimation results increases significantly when avoiding both suboptimal solutions and overfitting. Suboptimal solutions can be excluded when using deterministic global optimization (DGO) methods [1, 2]. Common DGO methods [3, 4] like the branch-and-bound (B&B) algorithm [5, 6] reach their limitations when solving large-scale nonconvex optimization problems arising from many parameter estimation problems [7–9].

To overcome these limitations, we [10] extended the standard B&B algorithm by using growing datasets. Briefly speaking, we start the B&B algorithm with a reduced model obtained by picking only a small subset of the full dataset and augment the dataset until converging to the full dataset provided. For this, we introduced *augmentation rules* which decide for each processed node whether to augment the dataset or to branch the parameter domain. We have proven that the B&B algorithm with growing datasets remains a DGO method, see Theorem 2 of [10]. A key point of the proof is that the lower bounding problem based on reduced datasets yields a valid lower bound for the *original problem*, i.e., the parameter estimation problem based on the full dataset. Another key point is the choice of an augmentation rule which guarantees reaching the full dataset eventually. However, when data reduction allows for a much looser lower bound and objective value, convergence may be slowed down or even prevented depending on the augmentation rule, see Example 1 in [10]. This brings us back to the topic of overfitting.

In the context of machine learning, overfitting refers to the phenomenon of models performing well on the data used for training but fail to generalize to new data [11, 12]. In practice, the actual performance of the trained model is therefore commonly estimated based on independent datasets, the so-called validation sets. We transfer these findings to the B&B algorithm with growing datasets by using out-of-sample evaluations for improving the lower bounds calculated based on the reduced dataset (*reduced lower bound*). In detail, we use an approximated lower bound given by a combination of the reduced lower bound and the out-of-sample evaluation for (i) a novel augmentation rule and (ii) for pruning. While (i) yields a heuristic rule aiming to improve the performance of the deterministic approach, with (ii) we loose the theoretical guarantee for converging towards the global solution and therefore obtain mere heuristic approaches. To estimate whether the heuristic pruning indeed cuts off the global solution, we propose a post-processing check of the final lower bound after the termination of the B&B algorithm.

In [10], we proposed to pick the data points for the reduced datasets randomly from the full dataset. In this article, we exploit this randomness for improving the reduced lower bound for both the heuristic augmentation rule and the heuristic pruning. In fact, resampling is a common tool in statistics to reduce the bias and variance of estimators, cf. [13, 14] and [15, Chapter 5]. In our approach, resampling a reduced dataset comes with the calculation and optimization of a lower bounding problem, making it computationally costly. We therefore propose to use two subsamples for updating the reduced lower bound based on the initial dataset.

All proposed extensions of the B&B algorithm with growing datasets are implemented in our open-source solver MAiNGO[1] [16] which is a DGO solver for factorable mixed-integer nonlinear programs. We perform an extensive case study comparing the deterministic and heuristic approaches of the B&B algorithm with growing datasets with the standard B&B algorithm as a benchmark. For this, we revisit real-world applications from our previous work and collected further models from literature resulting in 13 different parameter

[1] Available at https://git.rwth-aachen.de/avt-svt/public/maingo.

estimation problems covering the fields of process systems engineering, biochemistry, and machine learning. Ready-to-use implementations of the problem formulations are published in our open-source repository GloPSE[2].

The remainder of this article is structured as follows. Section 2 comprises the mathematical background, where we start with an overview on our notation in Section 2.1. Subsequently, the main algorithmic and theoretical advances of the B&B algorithm with growing datasets from [10] are recalled in Section 2.2. In Section 3, we obtain an approximate lower bound for the original problem based on the reduced lower bound and out-of-sample evaluation. This approximation is used for a novel augmentation rule in Section 3.1, and for introducing a heuristic approach including a post-processing step in Section 3.2. In Section 3.3, we transfer the findings to estimation problems minimizing the mean squared error yielding a second heuristic approach. In Section 4, we propose a heuristic lower bound based on resampling. In Section 5, we perform an extensive case study evaluating the proposed post-processing procedure and comparing all approaches of the B&B algorithm with growing datasets with the standard B&B algorithm, before we conclude in Section 6.

2 Preliminaries

2.1 Problem formulation and notation

As in our previous work [10], we focus on finding globally optimum parameter values $\boldsymbol{p} \in \mathcal{P}$ minimizing

$$\begin{aligned} \min_{\boldsymbol{p}\in\mathcal{P}} \quad & \sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}} g(\boldsymbol{p}; \boldsymbol{x}_d, y_d) && \text{(SSE)} \\ \text{s.t. } & h(\boldsymbol{x}_d, y_d; \boldsymbol{p}) \leq 0 \qquad \forall (\boldsymbol{x}_d, y_d) \in \mathcal{D} \\ & \widetilde{h}(\boldsymbol{p}) \leq 0, && \text{(PE)} \end{aligned}$$

where $g(\cdot; \boldsymbol{x}_d, y_d) = (f(\boldsymbol{x}_d; \cdot) - y_d)^2$ is the squared prediction error of a parameterized model function $f(\cdot; \boldsymbol{p}) : \mathbb{R}^m \to \mathbb{R}$ for data point $(\boldsymbol{x}_d, y_d) \in \mathcal{D} \subsetneq \mathbb{R}^m \times \mathbb{R}$, and $h(\boldsymbol{x}_d, y_d; \cdot)$ and $\widetilde{h}$ are the residual functions of the data-dependent and data-independent inequality constraints on parameters $\boldsymbol{p} \in \mathcal{P}$, respectively. Let the parameter domain $\mathcal{P} \subsetneq \mathbb{R}^n$ be a closed, bounded box. In case of integer parameters, let the parameter domain consist of subsequent discrete values. Note that $g(\cdot; \boldsymbol{x}_d, y_d)$ is nonnegative, and strictly positive in case of model-data mismatch caused by measurement errors, suboptimal parameter values, or model misspecification.

In the following, we call $\mathcal{D}$ the full dataset and parameter estimation problem (PE) the *original problem*. Let $g^{\text{cv}}(\cdot; \boldsymbol{x}_d, y_d) : \mathcal{P} \to \mathbb{R}$ be a non-negative convex underestimator of $g(\cdot; \boldsymbol{x}_d, y_d)$ as well as $h^{\text{cv}}(\boldsymbol{x}_d, y_d; \cdot)$ and $\widetilde{h}^{\text{cv}}$ be convex underestimators of $h(\boldsymbol{x}_d, y_d; \cdot)$ and $\widetilde{h}$, respectively. Then, the optimal solution of the convex optimization problem

$$\begin{aligned} \min_{\boldsymbol{p}\in\mathcal{P}} \quad & \sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}} g^{\text{cv}}(\boldsymbol{p}; \boldsymbol{x}_d, y_d) && \text{(LBP)} \\ \text{s.t. } & h^{\text{cv}}(\boldsymbol{x}_d, y_d; \boldsymbol{p}) \leq 0 \qquad \forall (\boldsymbol{x}_d, y_d) \in \mathcal{D} \\ & \widetilde{h}^{\text{cv}}(\boldsymbol{p}) \leq 0\,, \end{aligned}$$

[2] Available at https://git.rwth-aachen.de/avt-svt/public/glopse.

gives a lower bound on the optimal solution of the original problem, which we will call *full lower bound*. We refer to *reduced problem* and *reduced lower bound* when replacing the dataset with a reduced dataset $\mathcal{D}_k \subseteq \mathcal{D}$ within (PE) and (LBP), respectively. Note that, in common DGO software, the lower bounding problem (LBP) is further relaxed to a linear program without losing its validity for the original problem by linearizing valid convex underestimators.

In the B&B node $N_k = (\mathcal{P}_k, \mathcal{D}_k)$ processed in B&B iteration $k \in \mathbb{N}$, optimization problems (PE) and (LBP) are solved for parameter domain $\mathcal{P}_k \subseteq \mathcal{P}$ and dataset $\mathcal{D}_k \subseteq \mathcal{D}$. Let u_k be the best upper bound found until iteration k. Let l_k^{full} and l_k^{red} be the full and reduced lower bound calculated in node N_k with optimal parameter values $\boldsymbol{p}_k^{\text{lb,full}}$ and $\boldsymbol{p}_k^{\text{lb,red}}$, respectively. If a lower bound calculated based on a reduced dataset cannot be guaranteed to be valid for the original problem, we will call it *heuristic lower bound* $\widehat{l}_k$ compared to valid lower bounds l_k.

2.2 The B&B algorithm with growing datasets

As proposed in [10], the B&B algorithm with growing datasets extends the standard algorithm by a data reduction step before entering the B&B loop and a subroutine for deciding whether to branch or augment a node, see Figure 1. For this, we associate all nodes $N_k = (\mathcal{P}_k, \mathcal{D}_k) \in \mathcal{N}$ of the B&B tree with both a parameter domain $\mathcal{P}_k \subsetneq \mathcal{P}$ and a dataset $\mathcal{D}_k \subseteq \mathcal{D}$. In the subroutine, cf. Subroutine 1 of [10], we call an augmentation rule $\mathbb{A} : \mathcal{N} \to \{\text{True, False}\}$. If $\mathbb{A}(N_k) =$True, the dataset is augmented, i.e., we add one child node $N^{\text{new}} = (\mathcal{P}_k, \mathcal{D}_{\text{new}})$ with $\mathcal{D}_{\text{new}} \supsetneq \mathcal{D}_k$. Otherwise, we branch the parameter domain while retaining the dataset, i.e., we add two child nodes $N^{\text{new},1} = (\mathcal{P}_{k,1}, \mathcal{D}_k)$ and $N^{\text{new},2} = (\mathcal{P}_{k,2}, \mathcal{D}_k)$ with partition $\mathcal{P}_{k,1} \cup \mathcal{P}_{k,2} = \mathcal{P}_k$. With this, we obtain an algorithm which is guaranteed to converge towards the global optimum of (PE) if we use a finitely convergent augmentation rule [10].

Definition 1 (Definition 1 of [10]) Let $\mathcal{D}_k \subseteq \mathcal{D}$ be the dataset used in node N_k, which is processed in iteration k of the B&B algorithm with growing datasets depicted in Figure 1. The augmentation rule $\mathbb{A} : \mathcal{N} \to \{\text{True, False}\}$ *completes finitely*, if for any infinite nested sequence of nodes $\{N_{k_j}\}_{j \to \infty}$ with $N_{k_j} = (\mathcal{P}_{k_j}, \mathcal{D}_{k_j})$ it holds $\exists J < \infty : \mathcal{D}_{k_j} = \mathcal{D}\ \forall j \geq J$.

For completeness of contents, we repeat the main theoretical results of [10]. The interested reader may refer to the original publication [10] for more details and the proves. The theoretical results are based on the following assumptions.

Assumption 1 (Assumptions 1 and 2 of [10])

(i) Let $f(\boldsymbol{x}_d; \cdot) : \mathcal{P} \to \mathbb{R}$ and $h(\boldsymbol{x}_d, y_d; \cdot) : \mathcal{P} \to \mathbb{R}$ for any fixed $(\boldsymbol{x}_d, y_d) \in \mathcal{D}$ as well as $\widetilde{h} : \mathcal{P} \to \mathbb{R}$ be continuous.
(ii) Let $g^{\text{cv}}(\cdot; \boldsymbol{x}_d, y_d)$ be any nonnegative convex underestimator of $g(\cdot; \boldsymbol{x}_d, y_d)$ over $\mathcal{P}$ for any fixed $(\boldsymbol{x}_d, y_d) \in \mathcal{D}$.
(iii) Let $h^{\text{cv}}(\boldsymbol{x}_d, y_d; \cdot)$ and $\widetilde{h}^{\text{cv}}$ be any convex underestimator of $h(\boldsymbol{x}_d, y_d; \cdot)$ for any fixed $(\boldsymbol{x}_d, y_d) \in \mathcal{D}$ and $\widetilde{h}$, respectively, over $\mathcal{P}$.

With the help of the nonnegative convex underestimators, cf. Assumption 1(ii), we can construct valid lower bounds based on a reduced dataset.

Lemma 1 (Lemma 1 (i) of [10] (adapted)) *Let Assumption 1 hold. Then, $\sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}_{red}} g^{cv}(\boldsymbol{p}; \boldsymbol{x}_d, y_d)$ is a convex underestimator of both $\sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}_{red}} g(\boldsymbol{p}; \boldsymbol{x}_d, y_d)$ and $\sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} g(\boldsymbol{p}; \boldsymbol{x}_d, y_d)$ over $\mathcal{P}$ for any $\mathcal{D}_{red} \subseteq \mathcal{D}$.*

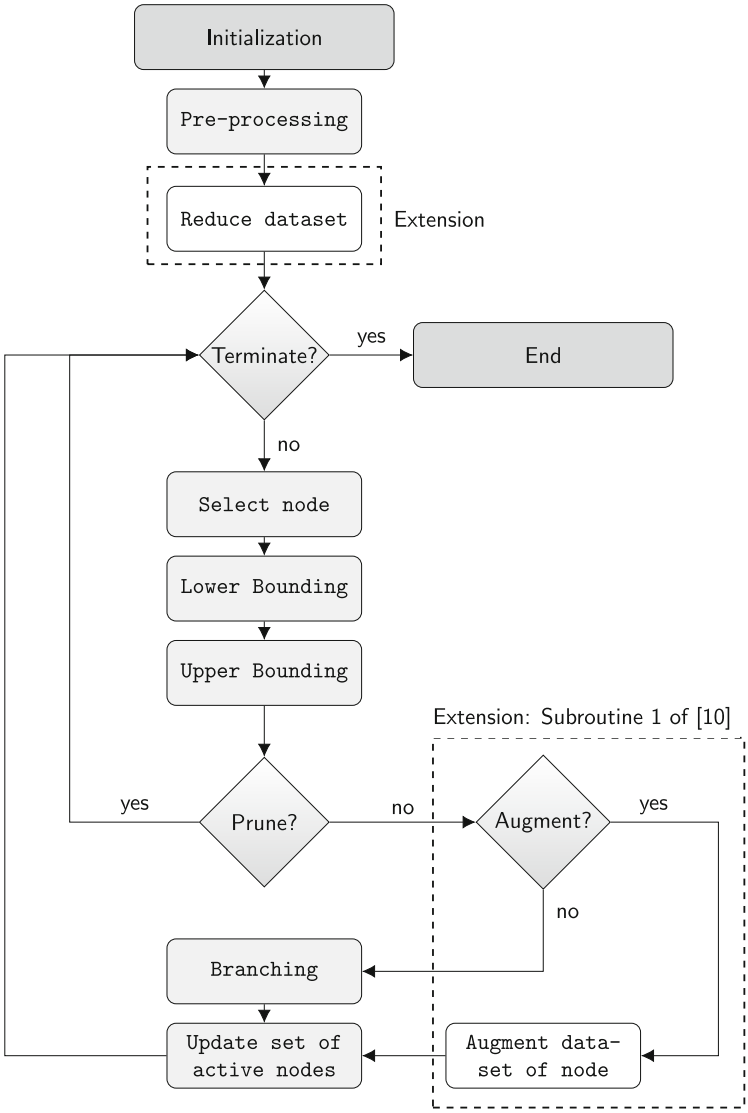

Fig. 1 Flow chart of the B&B algorithm with growing datasets [10] where the extensions of the standard B&B algorithm are highlighted with dashed boxes.

Finally, we extended the proof of convergence of a standard B&B algorithm given in Theorem 5.26 of Locatelli and Schoen [17]. In particular, we build upon their definitions for *exactness in the limit*, the *isotonic property* of underestimators, and *exhaustiveness* of branching given in Definition 5.4, Equation (5.34), and Definition 5.5, respectively, of [17].

Theorem 1 (Theorem 2 of [10] (adapted)) *Let Assumption 1 hold. We apply a spatial B&B algorithm with optimality tolerance* $\varepsilon > 0$ *to (PE). Let the convex underestimators* $g^{cv}_{\mathcal{D}}$, $h^{cv}(\boldsymbol{x}_d, y_d; \cdot)\ \forall (\boldsymbol{x}_d, y_d) \in \mathcal{D}$, *and* $\widetilde{h}^{cv}$ *be exact in the limit. Let* $g^{cv}(\cdot; \boldsymbol{x}_d, y_d)$ *and*

Table 1 Datasets, corresponding optimal objective and scaled optimal objective as given for Example 1 of [10]

Data points	Opt. obj.	Opt. obj. scaled with $\frac{\lvert\mathcal{D}\rvert}{\lvert\mathcal{D}_{\text{red}}\rvert}$
$\mathcal{D}_0 = \{(0, 0.6)\}$	0.0	$3 \cdot 0 = 0$
$\mathcal{D}_1 = \{(0, 0.6), (0, 1)\}$	0.08	$\frac{3}{2} \cdot 0.08 = 0.12$
$\mathcal{D} = \{(0, 0), (0, 0.6), (0, 1)\}$	0.5067	0.5067

$h^{cv}(\boldsymbol{x}_d, y_d; \cdot)$ satisfy the isotonic property $\forall(\boldsymbol{x}_d, y_d) \in \mathcal{D}$. Let the subdivision process of the B&B algorithm be exhaustive.
If we use an augmentation rule $\mathbb{A}$ which completes finitely, then the B&B algorithm with growing datasets depicted in Figure 1 terminates after a finite number of iterations and

- *either establishes that the problem is infeasible if the final upper bound equals infinity*
- *or returns an ε-optimal solution if the final lower bound is finite.*

Note that it is essential for the proof of *finite* convergence that the augmentation rule completes finitely. Otherwise there may be a gap between reduced and full lower bound preventing augmenting or even pruning based on the reduced dataset, see the following example.

Example 1 (Example 1 of [10] (adapted)) We want to solve

$$\min_{a \in [0,25]} (a-1)^2 + (a-0.6)^2 + (a-0)^2 \,,$$

where we find a linear function $f(x; a) = a \cdot x$ through the origin and three data points $\mathcal{D} = \{(1, 0), (1, 0.6), (1, 1)\}$ at the same input $x = 1$, with a naive implementation of the B&B algorithm with growing datasets using augmentation rule $\mathbb{A}_{\text{SCALING}}$ with $\rho = 1$. Assume that the (reduced) datasets are chosen as in Table 1. Even if the lower bound calculated based on reduced datasets $\mathcal{D}_0$ and $\mathcal{D}_1$ is exact, i.e., equals the respective optimal objective at the optimum point $\boldsymbol{p}^*$, gaps would remain

$$\begin{aligned}\sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}_0} g^{\text{cv}}(\boldsymbol{p}^*; \boldsymbol{x}_d, y_d) &\ll \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}_1} g^{\text{cv}}(\boldsymbol{p}^*; \boldsymbol{x}_d, y_d) \\ &\ll \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} g^{\text{cv}}(\boldsymbol{p}^*; \boldsymbol{x}_d, y_d) \leq \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} g(\boldsymbol{p}^*; \boldsymbol{x}_d, y_d)\end{aligned}$$

Branching does not help either since we already assume exact lower bounds. With that, neither augmenting nor convergence is possible. □

3 Out-of-sample estimation with growing datasets

The reduced lower bound is valid for the original problem but may be too loose for augmenting or pruning, cf. Section 2.2 and [10]. In this case, the time savings due to the data reduction may be canceled out by the unnecessary large B&B tree. Thus, we propose to apply out-of-sample evaluation for a better approximation of the full lower bound. We expect that the solution of the reduced bounding problem plus this additional evaluation remains computationally faster

than calculating the full lower bound. In detail, we evaluate (LBP) over all remaining data points $\mathcal{D} \setminus \mathcal{D}_k$ at the optimal solution point $\boldsymbol{p}_k^{\text{lb,red}}$ yielding the heuristic *out-of-sample lower bound*

$$\widehat{l}_k^{\text{oos}} := \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D} \setminus \mathcal{D}_k} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d) \ .$$

Note that we define the out-of-sample lower bound only for proper subsets $\mathcal{D}_k \subsetneq \mathcal{D}$. Based on empirical evidence, we expect the out-of-sample lower bound to be valid for the original problem. In fact, the out-of-sample lower bound sums up less nonnegative terms than the full lower bound, where each term is a function evaluation at point $\boldsymbol{p}_k^{\text{lb,red}}$ which is typically close or equal to the optimal solution point $\boldsymbol{p}_k^{\text{lb,full}}$.

Postulate 1 *For B&B nodes with reduced datasets $\mathcal{D}_k \subsetneq \mathcal{D}$, out-of-sample lower bound $\widehat{l}_k^{oos}$ is smaller than or equal the full lower bound l_k^{full} and therefore valid for the original problem.*

Postulate 1 may be violated for pathological cases as shown in the following example.

Example 2 (Non-valid $\widehat{l}_k^{\text{oos}}$) Assume that we are looking for the best slope of a linear function through data points $\mathcal{D} = \{(1, 1), (2, 5.5), (3, 3)\} \subsetneq \mathbb{R}^2$ yielding the unconstrained convex parameter estimation problem

$$\min_{p \in [0,10]} \sum_{(x_d, y_d) \in \mathcal{D}} (p \cdot x_d - y_d)^2 \quad . \tag{1}$$

Assume further that the reduced dataset at B&B iteration k is given by $\mathcal{D}_k = \{(1, 1), (3, 3)\}$. Due to the convexity of (1), we can solve it also as the lower bounding problem. We observe $l_k^{\text{red}} = 0$ at optimal solution point $p_k^{\text{lb,red}} = 1$, meaning that we can fit data points $\mathcal{D}_k$ exactly by the identity. We obtain $\widehat{l}_k^{\text{oos}} = (1 \cdot 2 - 5.5)^2 = 12.25$, and $l_k^{\text{full}} = 8.75$ at optimal solution point $p_k^{\text{lb,red}} = 1.5$. Hence, out-of-sample lower bound $\widehat{l}_k^{\text{oos}}$ is larger than the full lower bound l_k^{full}.

With the help of both reduced and out-of-sample lower bound, we can enclose the full lower bound.

Lemma 2 *Let $g^{cv}(\cdot; \boldsymbol{x}_d, y_d)$ be Lipschitz continuous in domain $\mathcal{P}$ with Lipschitz constants $L_d > 0$ for any $d = 1, \ldots, |\mathcal{D}|$. If $\boldsymbol{p}_k^{lb,red}$ is feasible for (LBP), then*

$$l_k^{red} + \widehat{l}_k^{oos} - L \cdot ||\boldsymbol{p}_k^{lb,full} - \boldsymbol{p}_k^{lb,red}|| \leq l_k^{full} \leq l_k^{red} + \widehat{l}_k^{oos}$$

with $L := \sum_{d=1}^{|\mathcal{D}|} L_d$.

Proof For the first inequality, we observe

$$\begin{aligned}
&\sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,full}}; \boldsymbol{x}_d, y_d) \\
&\quad = \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} \left(g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d) + g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,full}}; \boldsymbol{x}_d, y_d) - g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d) \right) \\
&\quad = \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}_k} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d) + \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D} \setminus \mathcal{D}_k} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d) \\
&\qquad + \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} \left(g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,full}}; \boldsymbol{x}_d, y_d) - g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d) \right) \ .
\end{aligned} \tag{2}$$

Since $\boldsymbol{p}_k^{\text{lb,red}}$ is feasible for (LBP) and $\boldsymbol{p}_k^{\text{lb,full}}$ minimizes (LBP), we have

$$\sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,full}}; \boldsymbol{x}_d, y_d) \leq \sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d)$$

and, together with Lipschitz continuity,

$$\begin{aligned} -\sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}} & \left(g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,full}}; \boldsymbol{x}_d, y_d) - g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d)\right) \\ &= \left|\sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}} \left(g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,full}}; \boldsymbol{x}_d, y_d) - g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d)\right)\right| \\ &\overset{\text{additivity}}{\leq} \sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}} \left|g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,full}}; \boldsymbol{x}_d, y_d) - g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d)\right| \\ &\overset{\text{Lipschitz}}{\leq} \sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}} L_d \cdot ||\boldsymbol{p}_k^{\text{lb,full}} - \boldsymbol{p}_k^{\text{lb,red}}||. \end{aligned} \tag{3}$$

Inserting (3) in (2) gives

$$\begin{aligned} \sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,full}}; \boldsymbol{x}_d, y_d) \geq & \sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}_k} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d) + \sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}\setminus\mathcal{D}_k} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d) \\ & - L \cdot ||\boldsymbol{p}_k^{\text{lb,full}} - \boldsymbol{p}_k^{\text{lb,red}}|| \end{aligned}$$

which is the same as the first inequality of Lemma 2.

For the second inequality, we observe

$$l_k^{\text{full}} = \sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,full}}; \boldsymbol{x}_d, y_d) \leq \sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d)$$

due to the optimality of $\boldsymbol{p}_k^{\text{lb,full}}$ and the feasibility of $\boldsymbol{p}_k^{\text{lb,red}}$ for the full lower bounding problem (LBP). Moreover, we have

$$\begin{aligned} \sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d) = & \sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}_k} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d) \\ & + \sum_{(\boldsymbol{x}_d, y_d)\in\mathcal{D}\setminus\mathcal{D}_k} g^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}}; \boldsymbol{x}_d, y_d) \end{aligned}$$

which concludes the proof. □

Regarding the feasibility of point $\boldsymbol{p}_k^{\text{lb,red}}$ for (LBP), we note that the data-independent constraints $\widetilde{h}^{\text{cv}}$ are invariant to the dataset used. Thus, the feasibility is naturally given for any model without data-dependent constraints $h^{\text{cv}}(\boldsymbol{x}_d, y_d; \cdot)$.

Based on Lemma 2, we define the heuristic *combined lower bound*

$$\widehat{l}_k^{\text{combi}} := l_k^{\text{red}} + \widehat{l}_k^{\text{oos}} \quad .$$

Note that the positivity of the cumulated Lipschitz constant L is crucial for bounding the full lower bound below based on the combined lower bound. Due to the exhaustiveness of branching, we have $\text{diam}(\mathcal{P}_k) \to 0$ for B&B iteration $k \to \infty$ and, thus, $||\boldsymbol{p}_k^{\text{lb,full}} - \boldsymbol{p}_k^{\text{lb,red}}|| \to 0$ $(k \to \infty)$. Even at the beginning of the B&B algorithm, i.e., for small k, the optimal points $\boldsymbol{p}_k^{\text{lb,full}}$ and $\boldsymbol{p}_k^{\text{lb,red}}$ often coincided in numerical experiments performed with the models investigated in Section 5. Consequently, we expect numerical advantages when

replacing the full lower bound with the combined lower bound as formalized in the following postulates.

Postulate 2 *1. For most B&B nodes, the combined lower bound $\widehat{l}_k^{combi}$ is (almost) equal to the full lower bound l_k^{full}.*
2. The evaluation of a lower bounding problem requires significantly smaller numerical effort than its optimization. In particular, calculating the combined lower bound $\widehat{l}_k^{combi}$ is numerically cheaper than calculating the full lower bound l_k^{full}.

In other words, we obtain a tight approximation of the full lower bound whose computation is still less costly compared to calculating the full dataset.

3.1 Deterministic approach

In the B&B algorithm with growing datasets, we either branch the parameter domain or augment the dataset of each B&B node which is not pruned, cf. Figure 1. As recalled in Section 2.2, this decision is made by so-called *augmentation rules*. In [10], we introduced the intuitive augmentation rule $\mathbb{A}_{\text{CONST}}$ triggering augmentation for nodes at a specific depth within the B&B tree

$$\mathbb{A}_{\text{CONST}}(N_k) := \begin{cases} \text{True}, & \text{if } \frac{\text{depth}(N_k)}{c} \in \mathbb{Z} \\ \text{False}, & \text{else} \end{cases}$$

with a user-defined constant $c \in \mathbb{Z}$. As an alternative, we proposed augmentation rule

$$\mathbb{A}_{\text{SCALING}}(N) := \begin{cases} \text{True}, & \text{if } \rho \cdot \widehat{l}_k^{\text{scaled}} \geq u_k - \varepsilon \\ \text{False}, & \text{else} \end{cases},$$

where u_k is the upper bound computed for the currently active node N_k, ρ a constant in $(0, 1]$ provided by the user, and $\widehat{l}_k^{\text{scaled}}$ the heuristic *scaled lower bound*

$$\widehat{l}_k^{\text{scaled}} := \frac{|\mathcal{D}|}{|\mathcal{D}_{\text{red}}|} \cdot l_k^{\text{red}} \quad .$$

Thus, $\mathbb{A}_{\text{SCALING}}$ with default augmentation weight $\rho = 1$ triggers augmentation for any node which could have been pruned based on the full lower bound under the expectation that the full lower bound equals the scaled lower bound. Based on Lemma 2, we assume that the combined lower bound gives a tighter approximation of the full lower bound than the scaled lower bound. We therefore propose augmentation rule COMBI

$$\mathbb{A}_{\text{COMBI}}(N_k) := \begin{cases} \text{True}, & \text{if } \widehat{l}_k^{\text{combi}} \geq u_k - \varepsilon \\ \text{False}, & \text{else} \end{cases},$$

where ε is the optimality tolerance. When using heuristic lower bounds solely within the augmentation rule and the valid lower bound l_k^{red} for pruning, we obtain a deterministic approach [10].

3.2 SSE heuristic

The downside of the deterministic approach is that the gap between reduced and full lower bound may slow down or even prevent convergence when using heuristic augmentation

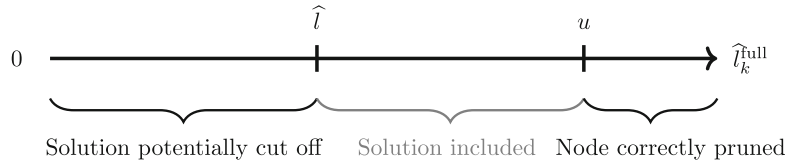

Fig. 2 Consequences for the optimal solution in dependence of the relation between post-processed lower bound l_k^{full} and both final lower bound $\widehat{l}$ and final upper bound u reported after termination of the B&B algorithm

rules like $\mathbb{A}_{\text{SCALING}}$ which do not complete finitely, compare Example 1. In theory, we can guarantee convergence when combining both rules to a hybrid augmentation rule $\mathbb{A}_{\text{SCALCST}} := \mathbb{A}_{\text{SCALING}} \vee \mathbb{A}_{\text{CONST}}$ [10]. However, this hybrid augmentation rule coincides with $\mathbb{A}_{\text{CONST}}$ in cases similar to the pathological problem in Example 1. Motivated by Lemma 2 and Postulate 2, we rather propose the *SSE heuristic* which uses the combined lower bound for *pruning* nodes in the B&B algorithm with growing datasets. Even when satisfying Postulate 2, the combined lower bound may be strictly larger than the full lower bound and may allow for pruning nodes containing the global solution of the original problem. Indeed, we may make a mistake for any node pruned based on a reduced dataset

$$\mathcal{N}^{\text{postpro}} := \{N_k : \widehat{l}_k^{\text{combi}} \geq u_k \text{ and } \mathcal{D}_k \subsetneq \mathcal{D}\} \quad .$$

As a post-processing step, we therefore use the *full* lower bound l_k^{full} of each node $N_k \in \mathcal{N}^{\text{postpro}}$ to update the final lower bound.

For post-processing, we need to distinguish three cases in dependence of the heuristic final lower bound $\widehat{l}$ and final upper bound u reported after termination of the B&B algorithm, see Figure 2. Firstly, it was correct to prune the node if its full lower bound satisfies $l_k^{\text{full}} > u$. Secondly, if l_k^{full} is within the final lower and upper bound reported, the global solution of the original problem lies within the user-given optimality tolerance no matter whether it is contained in node N_k or another node. Thirdly, if l_k^{full} is smaller than the (heuristic) final lower bound $\widehat{l}$, we may have cut off the global solution. In this case, we know $l_k^{\text{full}} < \widehat{l} \leq u \leq u_k \leq \widehat{l}_k^{\text{combi}}$. To obtain a deterministic approach, we would need to re-run the B&B algorithm for all nodes

$$\mathcal{N}^{\text{changed}} := \{N_k \in \mathcal{N}^{\text{postpro}} : l_k^{\text{full}} < \widehat{l}\}$$

as the root node. However, this can become very costly, especially if the parameter domains of the respective nodes are comparatively large. Instead we update the final lower bound by the minimum over $l_k^{\text{full}},\ k : N_k \in \mathcal{N}^{\text{changed}}$ yielding a more accurate value for the true lower bound on the optimal solution, see Subroutine 1. To limit the numerical effort for post-processing, we only track the 100 nodes of $\mathcal{N}^{\text{postpro}}$ with the smallest values for $\widehat{l}_k^{\text{combi}}$, i.e., the 100 tightest pruning decisions, see Subroutine 2.

We can transfer augmentation rules $\mathbb{A}_{\text{CONST}}$ and $\mathbb{A}_{\text{SCALING}}$ from the deterministic approach to the SSE heuristic without any changes, see Figure 3. However, the interval between combined and scaled lower bound may vanish if both are accurate approximations of the full lower bound. In that case, augmentation rule $\mathbb{A}_{\text{SCALING}}$ cannot trigger augmentation. We therefore introduce an interval of uncertainty around the combined lower bound used for pruning, and augment if we fall within this tolerance $\widehat{\varepsilon}$ giving augmentation rule TOL

$$\mathbb{A}_{\text{TOL}}(N_k) := \begin{cases} \text{True}, & \text{if } \widehat{l}_k^{\text{combi}} \geq u_k - \widehat{\varepsilon} \\ \text{False}, & \text{else} \end{cases} \quad .$$

Algorithm 1 Post-processing

Require: Set of nodes tracked for post-processing $\mathcal{N}^{\text{postpro}}$, final lower bound $\widehat{l}$ after termination of B&B algorithm
Ensure: Updated final lower bound $\widehat{l}$
1: **for all** $k : N_k \in \mathcal{N}^{\text{postpro}}$ **do**
2: Solve (LBP) over $\mathcal{P}_k$ based on full dataset $\mathcal{D}$ and obtain l_k^{full}
3: **if** $\widehat{l} > l_k^{\text{full}}$ **then**
4: $\widehat{l} := l_k^{\text{full}}$
5: **end if**
6: **end for**

Algorithm 2 Tracking nodes for post-processing within pruning step

Require: Set of nodes tracked for post-processing $\mathcal{N}^{\text{postpro}}$, currently pruned node N with lower bound $\widehat{l}^{\text{combi}}$
Ensure: Updated set $\mathcal{N}^{\text{postpro}}$
1: **if** $|\mathcal{N}^{\text{postpro}}| < 100$ **then**
2: $\mathcal{N}^{\text{postpro}} := \mathcal{N}^{\text{postpro}} \cup \{N\}$
3: **else**
4: $k_0 := \arg\max_{k:N_k \in \mathcal{N}^{\text{postpro}}} \{\widehat{l}_k^{\text{combi}}\}$
5: **if** $\widehat{l}^{\text{combi}} < \widehat{l}_{k_0}^{\text{combi}}$ **then**
6: $\mathcal{N}^{\text{postpro}} := \mathcal{N}^{\text{postpro}} \setminus \{N_{k_0}\} \cup \{N\}$
7: **end if**
8: **end if**

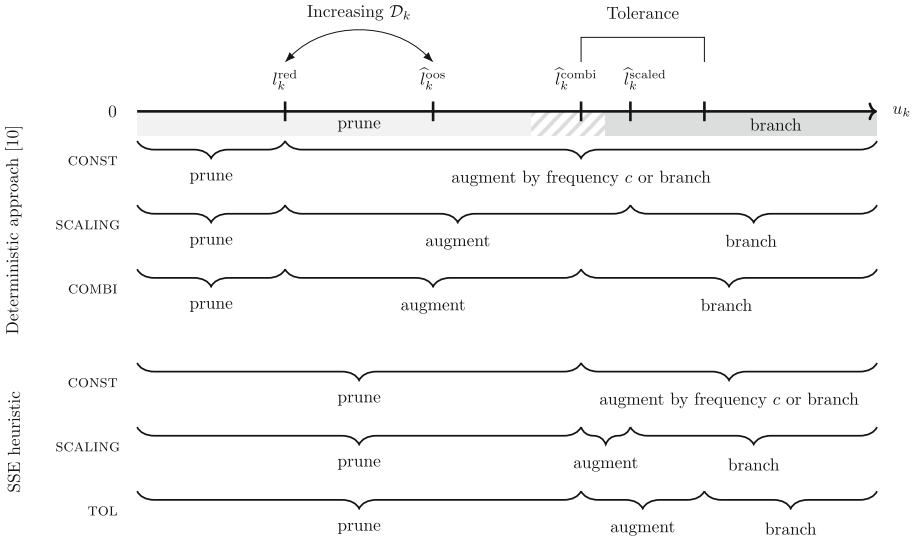

Fig. 3 Schematic overview of fathoming a node with objective (SSE) in dependence of its upper bound u_k: pruning, augmenting, and branching based on different augmentation rules as well as the postulated order of reduced, combined, and out-of-sample lower bound. The shaded rectangles indicate the exact fathoming based on the full dataset, where we expect the full lower bound to be in the the hatched region

In MAiNGO, we prune nodes after deducting the optimality tolerance, i.e., if $\widehat{l}_k^{\text{combi}} \geq u_k - \varepsilon$. To ensure a non-empty interval $[\widehat{l}_k^{\text{combi}} + \varepsilon, \widehat{l}_k^{\text{combi}} + \widehat{\varepsilon}]$ for triggering augmentation, we use ten times the default optimality tolerance ε for the augmentation tolerance $\widehat{\varepsilon}$, namely $\widehat{\varepsilon} = 0.1$ by default.

3.3 MSE heuristic

In estimation problems from literature, the mean squared error

$$\frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} g(\boldsymbol{p}; \boldsymbol{x}_d, y_d) \tag{MSE}$$

is preferred as objective over the summed squared error (SSE). Both objectives lead to the same optimal parameter values as they only differ by a factor independent of the parameters. However, the relations of reduced, full, and out-of-sample lower bound changes which may affect the convergence behavior of the B&B algorithm. For example, we expect the reduced lower bound to converge from below and the out-of-sample lower bound to converge from above to the full lower bound as is true for in-sample and out-of-sample estimation in machine learning during training [11]. We therefore propose the *MSE heuristic* on top of the SSE heuristic by replacing objective (SSE) with (MSE) in (PE).

Let $\boldsymbol{p}_{k,\text{MSE}}^{\text{lb,full}}$ and $\boldsymbol{p}_{k,\text{MSE}}^{\text{lb,red}}$ be the minimizer of (LBP) over domain $\mathcal{P}_k$ with objective function (MSE) based on dataset $\mathcal{D}$ and $\mathcal{D}_k$, respectively. We denote the respective full and heuristic reduced lower bound with $l_k^{\text{full,MSE}}$ and $\widehat{l}_k^{\text{red,MSE}}$. Note that the reduced lower bound may be larger than the full lower bound or even the optimal solution when using (MSE).

Additionally, we adapt the definition of the heuristic out-of-sample lower bound yielding

$$\widehat{l}_k^{\text{oos,MSE}} := \frac{1}{|\mathcal{D}| - |\mathcal{D}_k|} \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D} \setminus \mathcal{D}_k} g^{\text{cv}}(\boldsymbol{p}_{k,\text{MSE}}^{\text{lb,red}}; \boldsymbol{x}_d, y_d) \quad .$$

When using (MSE), the combined lower bound is given by a convex combination rather than the sum of reduced and out-of-sample lower bound.

Lemma 3 *Let $g^{cv}(\cdot; \boldsymbol{x}_d, y_d)$ be Lipschitz continuous in domain $\mathcal{P}$ with Lipschitz constants $L_d > 0$ for any $d = 1, \ldots, |\mathcal{D}|$. If $\boldsymbol{p}_{k,MSE}^{lb,red}$ is feasible for (LBP) with objective (MSE), then*

$$l_k^{full,MSE} \geq \; r_k \cdot \widehat{l}_k^{red,MSE} + (1 - r_k) \cdot \widehat{l}_k^{oos,MSE} - L^{MSE} \cdot ||\boldsymbol{p}_{k,MSE}^{lb,full} - \boldsymbol{p}_{k,MSE}^{lb,red}||$$

with $L^{MSE} := \frac{1}{|\mathcal{D}|} \sum_{d=1}^{|\mathcal{D}|} L_d$ and data ratio $r_k := \frac{|\mathcal{D}_k|}{|\mathcal{D}|}$.

Proof Analogously to the proof of Lemma 2, we obtain

$$\begin{aligned} \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} g^{\text{cv}}(\boldsymbol{p}_{k,\text{MSE}}^{\text{lb,full}}; \boldsymbol{x}_d, y_d) \geq \; & \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}_k} g^{\text{cv}}(\boldsymbol{p}_{k,\text{MSE}}^{\text{lb,red}}; \boldsymbol{x}_d, y_d) \\ & + \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D} \setminus \mathcal{D}_k} g^{\text{cv}}(\boldsymbol{p}_{k,\text{MSE}}^{\text{lb,red}}; \boldsymbol{x}_d, y_d) \\ & - \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} L_d \cdot ||\boldsymbol{p}_{k,\text{MSE}}^{\text{lb,full}} - \boldsymbol{p}_{k,\text{MSE}}^{\text{lb,red}}||. \end{aligned}$$

Observing $1 - r_k = \frac{|\mathcal{D}| - |\mathcal{D}_k|}{|\mathcal{D}|} = \frac{|\mathcal{D} \setminus \mathcal{D}_k|}{|\mathcal{D}|}$ concludes the proof. □

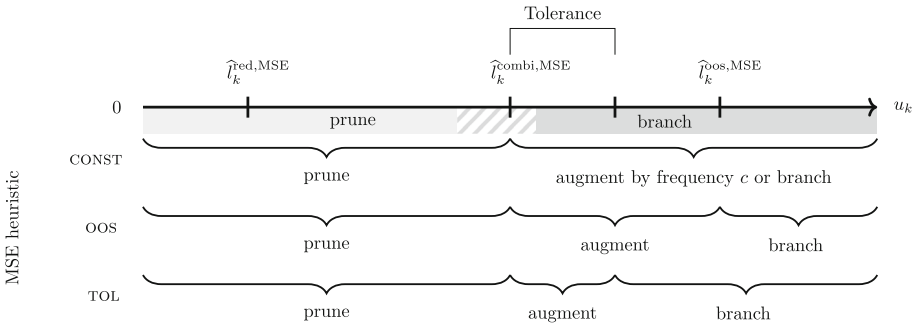

Fig. 4 Schematic overview of fathoming a node with objective (MSE) in dependence of its upper bound u_k: pruning, augmenting, and branching based on different augmentation rules as well as the postulated order of reduced, combined, and out-of-sample lower bound. The shaded rectangles indicate the exact fathoming based on the full dataset, where we expect the full lower bound to be in the the hatched region

Lemma 3 shows that we can expect the combined lower bound

$$\widehat{l}_k^{\text{combi,MSE}} := r_k \cdot \widehat{l}_k^{\text{red,MSE}} + (1 - r_k) \cdot \widehat{l}_k^{\text{oos,MSE}}$$

to be a valid lower bound of the original problem if the optimal solution points $\boldsymbol{p}_{k,\text{MSE}}^{\text{lb,full}}$ and $\boldsymbol{p}_{k,\text{MSE}}^{\text{lb,red}}$ coincide. The positive mean Lipschitz constant L^{MSE} allows for bounding the difference $\widehat{l}_k^{\text{combi,MSE}} - l_k^{\text{full,MSE}}$ from above. Again, this bound converges to 0 with increasing B&B iteration k due to the exhaustiveness of branching. Note that combined, reduced, and full lower bound coincide for $\mathcal{D}_k = \mathcal{D}$ by definition. Analogously to Postulate 2, we expect the following:

Postulate 3 *1. For most B&B nodes, the combined lower bound $\widehat{l}_k^{combi,MSE}$ is (almost) equal to the full lower bound $l_k^{full,MSE}$.*

2. Calculating the combined lower bound $\widehat{l}_k^{combi,MSE}$ is numerically cheaper than calculating the full lower bound $l_k^{full,MSE}$.

In the MSE heuristic, we therefore propose to use the combined lower bound $\widehat{l}_k^{\text{combi,MSE}}$ for pruning and to apply the post-processing procedure described in Subroutines 1 and 2 with the adapted definition of the combined and full lower bound.

We re-use augmentation rules $\mathbb{A}_{\text{CONST}}$ and $\mathbb{A}_{\text{TOL}}$ for the MSE heuristic, compare Figure 4. Since $\mathbb{A}_{\text{SCALING}}$ was motivated by using the reduced and full mean squared error, it does not add value when using (MSE). Instead, we introduce augmentation rule OOS

$$\mathbb{A}_{\text{OOS}}(N_k) := \begin{cases} \text{True}, & \text{if } \widehat{l}_k^{\text{oos,MSE}} \geq u_k^{\text{MSE}} - \varepsilon \\ \text{False}, & \text{else} \end{cases}$$

which triggers augmentation if we detect overfitting in form of a gap between the full lower bound, approximated by the combined lower bound, and the out-of-sample lower bound preventing pruning.

4 Resampling the initial dataset

In machine learning and statistics, resampling techniques like cross-validation, random forests, boosting, and bootstrap aggregation are used to reduce the estimation variance, see [15, Chapter 5 and Section 8.2] and [18, Chapters 7−10 and 15]. For example, the average generalization error of random forests containing many trees is proven to be at least as small as the average generalization error of one random tree, see Theorem 11.2 of [19]. We transfer this property to the aggregation of reduced lower bound using bootstrap aggregation or, in short, *bagging* [20]. In our approach, bagging means to randomly choose another reduced dataset $\mathcal{D}_{k,*}$, solve the lower bounding problem another time to obtain lower bound $g^{\mathrm{cv}}_{\mathcal{D}_{k,*}}(\boldsymbol{p}_k^{\mathrm{lb,red},*})$, and aggregate the resulting reduced lower bounds.

We observed in [10] that the computational effort for solving the lower bounding problem is approximately linear in the size of the used dataset. In other words, using two subsamples already means doubling the computational effort. We therefore restrict ourselves to a maximum of two subsamples. However, the results of Lemmas 4 and 5 derived in the following can be easily extended to the use of more subsamples.

We define

$$g_{\mathcal{D}_k}(\cdot) := \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}_k} g^{\mathrm{cv}}(\cdot; \boldsymbol{x}_d, y_d)$$

and quote the assumptions on the datasets from [10].

Assumption 2 (Assumption 3 of [10])

(i) The full dataset $\mathcal{D}$ is non-random, i.e., $\mathcal{D}$ is a fixed set and not considered as a random sample.
(ii) Let $\mathcal{D}_{\mathrm{red}} \subsetneq \mathcal{D}$ be a reduced dataset with an a-priori fixed size, where the data points in $\mathcal{D}_{\mathrm{red}}$ are picked randomly from $\mathcal{D}$ such that $\mathcal{D}_{\mathrm{red}}$ follows a discrete uniform probability distribution over all subsets of $\mathcal{D}$ with size $|\mathcal{D}_{\mathrm{red}}|$.

Based on this, we can show that the expected value EV $[\cdot]$ of the lower bound remains the same when applying bagging.

Lemma 4 *Let $\mathcal{D}_{k,1}$ and $\mathcal{D}_{k,2}$ be chosen according to Assumption 2. Then,*

$$\mathrm{EV}\left[\frac{g^{cv}_{\mathcal{D}_{k,1}}(\boldsymbol{p}_k^{lb,red,1}) + g^{cv}_{\mathcal{D}_{k,2}}(\boldsymbol{p}_k^{lb,red,2})}{2}\right] = \mathrm{EV}\left[g^{cv}_{\mathcal{D}_{k,1}}(\boldsymbol{p}_k^{lb,red,1})\right].$$

The statement of Lemma 4 follows from $\mathcal{D}_{k,1}$ and $\mathcal{D}_{k,2}$ having the same distribution and the linearity of the expected value.

In contrast to that, the variance of the aggregated reduced lower bound may be significantly smaller than the variance of a single reduced lower bound. In the following, Var $[\cdot]$ indicates the variance of a random variable, while Corr $[\cdot, \cdot]$ indicates Pearson's correlation coefficient between two random variables.

Lemma 5 *Let $\mathcal{D}_{k,1}$ and $\mathcal{D}_{k,2}$ be chosen according to Assumption 2.*

Then,

$$\begin{aligned}&\mathrm{Var}\left[\frac{g^{cv}_{\mathcal{D}_{k,1}}(\boldsymbol{p}_k^{lb,red,1})+g^{cv}_{\mathcal{D}_{k,2}}(\boldsymbol{p}_k^{lb,red,2})}{2}\right]\\&=\frac{1+\mathrm{Corr}\left[g^{cv}_{\mathcal{D}_{k,1}}(\boldsymbol{p}_k^{lb,red,1}),\, g^{cv}_{\mathcal{D}_{k,2}}(\boldsymbol{p}_k^{lb,red,2})\right]}{2}\cdot\mathrm{Var}\left[g^{cv}_{\mathcal{D}_{k,1}}(\boldsymbol{p}_k^{lb,red,1})\right].\end{aligned}$$

The statement of Lemma 5 follows from $\mathcal{D}_{k,1}$ and $\mathcal{D}_{k,2}$ having the same distribution as well as the scaling and additive properties of variances.

As $\mathrm{Corr}\,[\cdot,\cdot]\in[-1,1]$ by the Cauchy-Schwarz inequality, Lemma 5 implies that bagging can only improve the variance

$$\mathrm{Var}\left[\frac{g^{\mathrm{cv}}_{\mathcal{D}_{k,1}}(\boldsymbol{p}_k^{\mathrm{lb,red},1})+g^{\mathrm{cv}}_{\mathcal{D}_{k,2}}(\boldsymbol{p}_k^{\mathrm{lb,red},2})}{2}\right]\leq\mathrm{Var}\left[g^{\mathrm{cv}}_{\mathcal{D}_{k,1}}(\boldsymbol{p}_k^{\mathrm{lb,red},1})\right].$$

Equality can only be attained for perfectly correlated datasets giving $\mathrm{Corr}\left[g^{\mathrm{cv}}_{\mathcal{D}_{k,1}}(\boldsymbol{p}_k^{\mathrm{lb,red},1}),\, g^{\mathrm{cv}}_{\mathcal{D}_{k,2}}(\boldsymbol{p}_k^{\mathrm{lb,red},2})\right]=1$. Contrarily, for uncorrelated datasets, i.e., a correlation of 0, we can half the variance by picking a second subsample

$$\mathrm{Var}\left[\frac{g^{\mathrm{cv}}_{\mathcal{D}_{k,1}}(\boldsymbol{p}_k^{\mathrm{lb,red},1})+g^{\mathrm{cv}}_{\mathcal{D}_{k,2}}(\boldsymbol{p}_k^{\mathrm{lb,red},2})}{2}\right]=\frac{1}{2}\mathrm{Var}\left[g^{\mathrm{cv}}_{\mathcal{D}_{k,1}}(\boldsymbol{p}_k^{\mathrm{lb,red},1})\right].$$

The correlation equals zero for independent random variables. However, we cannot expect different reduced datasets to be independent, since (i) there may be systematic error affecting a subset of data points similarly and (ii) the datasets are drawn from the same sample $\mathcal{D}$ and may therefore contain common data points. The extent of (i) highly depends on the model and the actual dataset $\mathcal{D}$ which are both part of the fixed problem formulation in our setup. For (ii), we have higher chances that different subsamples $\mathcal{D}_{k,1}$ and $\mathcal{D}_{k,2}$ are not intersecting, if the size of the reduced datasets $\mathcal{D}_k$ is small compared to the full dataset $\mathcal{D}$. By default settings, we pick 10% of the data points from the full dataset for the initial dataset and add another 25% of the data points when augmenting. This means the second smallest dataset contains already a comparatively large part of the full dataset, namely 35% of all data points. For example, assume a full dataset with $|\mathcal{D}|=100$. The probability that two subsamples with $|\mathcal{D}_{k,*}|=10$ do not intersect is about 0.33, while it is about 2.75×10^{-9} for subsamples with $|\mathcal{D}_{k,*}|=35$. Thus, we use the proposed resampling heuristic only to update the reduced lower bound calculated based on the initial dataset.

5 Numerical results

In this section, we study the computational performance of the B&B algorithm with growing datasets using the standard B&B algorithm as a benchmark. In detail, we investigate the deterministic approach from [10] as well as the heuristic approaches from Sections 3.2 and 3.3 using different augmentation rules. We run each of the approaches with and without resampling the initial dataset, cf. Section 4. Both the standard B&B algorithm and all discussed algorithmic approaches of the B&B algorithm with growing datasets are available in our open-source solver MAiNGO v0.8.2.

Table 2 Overview on general properties and references of the models and data used in the case study

Name	n	$\lvert\mathcal{D}\rvert$	$\dim(\boldsymbol{x})$	Data	Opt. class	Original references Model	Data
Process systems engineering							
EOS262	9	262	2	synthetic, exact	MINLP	[21, 22]	[10]
EOS2262	9	2262	2	synthetic, exact	MINLP	— ” —	— ” —
EOS262noisy	9	262	2	synthetic, noisy	MINLP	— ” —	— ” —
EOS2262noisy	9	2262	2	synthetic, noisy	MINLP	— ” —	— ” —
IHMcon	20	284	1	measured	NLP	[23]	[24, 25]
IHMunc	10	284	1	measured	NLP	— ” —	— ” —
kinetics	5	446	1	measured	DO	[1, 2, 26]	[2]
EIS	7	26	1[a]	measured	NLP	[27]	[28, 29]
Biochemistry							
TSP	12	20	8	synthetic, exact	DO	[30, 31]	[10]
TSPnoisy	12	20	8	synthetic, noisy	DO	— ” —	— ” —
Machine learning							
GMMcon	6	272	1	measured	NLP	[32]	[33, 34]
GMMineq	5	272	1	measured	NLP	— ” —	— ” —
trainANN	21	220	3	measured	NLP	[35, 36]	[37, 38]

[a] The model output is a complex number, i.e., $\dim(y) = 2$

Benchmark libraries from mathematical programming, e.g., MINLPlib [39], PrincetonLib [40], and the COCONUT benchmark [41], only account for parameter estimation problems (PE) considering small datasets which do not require our extension. Besides, our focus is the solution of real-world applications extending the numerical proof-of-concept obtained in [10]. Hence, we collected different models from both literature and our previous work stemming from process systems engineering, machine learning, and biochemistry. These models cover different classes of optimization problems including mixed-integer nonlinear programs (MINLPs), nonlinear programs (NLPs), and dynamic optimization problems (DOs) with up to $n = 21$ unknown parameters and a different number and dimension of data points in full dataset $\mathcal{D}$, see Table 2. The exact mathematical expressions are provided in Online Resource 1. For ready-to-use implementations of the model as well as the exact data points used refer to the aforementioned repository GloPSE[2].

All computations are performed on Intel Xeon Platinum 8160 processors “SkyLake” (frequency 2.1GHz, RAM = 3.75GB for a single node). Even small deviations in computational times for processing one node may significantly change the number of nodes processed within our CPU time limit of 23h when adding up. Consequentially, the final lower and upper bound may change as well. Thus, we repeat each run 5 times and report the median of the final bounds and, in case of convergence, CPU times. Note that we choose a CPU time limit of 23h such that we can guarantee a total runtime of 24h including the initialization of the model and preparations of the output data.

We set the relative optimality tolerance to $\varepsilon^R = 0.1$. The absolute tolerance is fixed to $\varepsilon^A = 0.01$ for all approaches using (SSE). When using (MSE), the objective is scaled by the number of data points. In analogy, we scale the absolute optimality tolerance to model specific values $\varepsilon^A = 0.01/\lvert\mathcal{D}\rvert$ when using (MSE). In each optimization run, we use Ipopt version 3.12.12 [42] for running 3 local searches as a pre-processing step in the root node. Moreover,

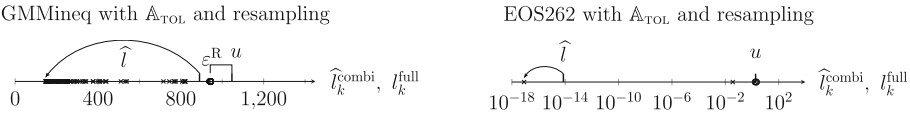

Fig. 5 Lower bounds of nodes in $\mathcal{N}^{\text{changed}}$ before (∘) and after (×) post-processing and change of final lower bound $\widehat{l}$ due to post-processing compared to final upper bound u and relative optimality tolerance ε^{R}

we use McCormick relaxations [43, 44] calculated with MC++ [45] and a linearization in the midpoint of the parameter intervals to obtain a linear program (LP) for lower bounding. For models IHMcon, IHMunc, EIS, TSP, TSPnoisy, GMMcon, GMMineq, and trainANN, we solve the LP with linear optimizer CLP v1.17.0 [46]. As the resulting LP of the EOS models with noisy measurement data as well as model kinetics seems to be numerically difficult for CLP and alternative LP solver CPLEX [47], we use interval extensions [48] calculated with FILIB++ [49] for models EOS262, EOS2262, EOS262noisy, EOS2262noisy, and kinetics. When using pure interval extensions, we disable the optimality-based bound tightening [50]. The upper bounding problem is solved with local optimizer LBFGS [51, 52] implemented in the NLOPT toolbox v2.5.0 [53] for all models. To minimize deviations caused by small differences in the CPU time required for processing a node, we limit the number of steps performed by local solvers Ipopt and LBFGS rather than the CPU time used for each local search. The complete listing of settings is provided in Online Resource 2.

5.1 Evaluation of post-processing

At first, we study the post-processing step to check whether we can expect the heuristic approaches to converge to the global solutions. Table 3 summarizes the number of nodes tracked for post-processing, the implications for the lower bounds reported, and the maximum CPU time used for post-processing. Note that these are the statistics of run 1 out of the 5 repetitive runs only. The number of nodes tracked as well as which specific nodes are tracked may vary if the number of processed nodes differs. However, we expect to make the largest mistakes early, namely with small datasets, which is covered by all 5 repetitive runs. More importantly, note that the statistics of models IHMcon, IHMunc, EIS, and trainANN are not listed, just as the statistics of the MSE heuristic for models TSP and TSPnoisy, since no nodes are tracked for post-processing in these cases. In these cases, we do not prune based on a reduced dataset. This means we have a deterministic procedure so far at the cost of a large B&B tree. Similarly, in runs with $|\mathcal{N}^{\text{postpro}}| < 100$ it seems to be hard to prune based on the reduced datasets. As an exception, $|\mathcal{N}^{\text{postpro}}| < 100$ comprises also the case where we need to process less than 100 nodes for convergence, cf. TSP model in Table 3.

We observe that for only 4 out of 13 models there are nodes where the lower bound l_k^{full} calculated in post-processing falls below the final upper bound, i.e., where we may have made a wrong pruning decision due to data reduction. Out of these, the final lower bound is updated solely for 3 models, namely when using the SSE heuristic applying augmentation rules SCALING and TOL for model EOS262 and augmentation rule TOL for models GMMcon and GMMineq.

We pick two exemplary cases to depict the changes due to post-processing in Figure 5. In both cases, the nodes which affect the final lower bound $\widehat{l}$ have been only just pruned with $\widehat{l}_k^{\text{combi}} \gtrsim (1 - \varepsilon^{\text{R}}) \cdot u$. In model EOS262, the changes due to post-processing are essentially nonexistent: we find numerically insignificant differences of 10^{-14} to 10^{-18}. In contrast to that, the final lower bound of models GMMcon and GMMineq change significantly, e.g.,

Table 3 Post-processing statistics for the models with tracked nodes, i.e., with $\mathcal{N}^{\text{postpro}} \neq \emptyset$, including the number of nodes tracked, the number of nodes where the lower bound changed significantly within post-processing, a flag indicating whether the final lower bound was changed as well, and the maximum CPU time over all augmentation rules

Model	Re-sampling?	$\lvert\mathcal{N}^{\text{postpro}}\rvert$	No. of nodes wrongly pruned[a]	Final lower bound changed?[b]	Maximum CPU time [s]
SSE heuristic using augmentation rules CONST/SCALING/TOL					
EOS262	No	100/100/100	0/2/2	0/1/1	0.7
	Yes	100/100/100	0/3/3	0/1/1	0.9
EOS2262	no	100/100/100	0/0/0	0/0/0	8.2
	Yes	100/100/100	0/0/0	0/0/0	9.9
EOS262noisy	no	22/72/72	0/0/0	0/0/0	0.6
	Yes	22/72/72	0/0/0	0/0/0	1.0
EOS2262noisy	no	52/100/100	0/0/0	0/0/0	7.1
	Yes	66/100/100	0/0/0	0/0/0	7.9
kinetics	no	100/100/100	0/0/0	0/0/0	1.6
	Yes	100/100/100	0/0/0	0/0/0	1.6
TSP	no	100/100/4	0/0/0	0/0/0	1.3
	Yes	100/100/4	0/0/0	0/0/0	0.3
TSPnoisy	no	16/100/0	0/0/0	0/0/0	0.2
	Yes	0/100/0	0/0/0	0/0/0	0.2
GMMcon	no	44/100/100	0/6/74	0/0/1	0.1
	yes	44/100/100	0/2/70	0/0/1	0.1

Table 3 continued

Model	Re-sampling?	$\lvert\mathcal{N}^{\text{postpro}}\rvert$	No. of nodes wrongly pruned[a]	Final lower bound changed?[b]	Maximum CPU time [s]
GMMineq	no	100/100/100	0/0/96	0/0/1	0.2
	yes	100/100/100	0/0/99	0/0/1	0.2
MSE heuristic using augmentation rules CONST/OOS/TOL					
EOS262	no	100/100/100	0/1/0	0/0/0	0.5
	yes	100/100/100	0/0/0	0/0/0	1.0
EOS2262	no	100/100/100	3/1/1	0/0/0	6.4
	yes	100/100/100	2/4/1	0/0/0	7.1
EOS262noisy	no	12/100/100	0/0/0	0/0/0	0.6
	yes	30/12/92	0/0/0	0/0/0	0.8
EOS2262noisy	no	4/76/76	0/0/0	0/0/0	5.2
	yes	12/76/76	0/0/0	0/0/0	5.8
kinetics	no	100/100/100	0/0/0	0/0/0	1.4
	yes	100/100/100	0/0/0	0/0/0	1.7
GMMcon	no	100/100/100	9/17/17	0/0/0	0.1
	yes	100/76/100	9/12/8	0/0/0	0.1
GMMineq	no	100/100/100	1/4/4	0/0/0	0.1
	yes	100/100/100	1/4/4	0/0/0	0.0

[a] Nodes with $l_k^{\text{full}} < u$, see Section 3.2

[b] 0 = no, 1 = yes

from 723.7 to 406.7 for GMMineq when using resampling, extending the relative optimality gap from 15% to 86%.

For 95% of 156 cases, including the 2 heuristic approaches with 3 different augmentation rules each for 13 models with and without resampling, the final lower bound was not changed and only in 4 cases the change was significant. In particular, the final lower bound was changed only when using the SSE heuristic. Note that the post-processing is very cheap: over all 5 repetitions of the case study we measured a maximum runtime of 11.3s.

In conclusion, even the heuristic approaches converge towards the global solution in most of the cases. The post-processing procedure detects the remaining cases with small computational effort and provides means for deciding whether the heuristic pruning significantly distorts the solution obtained.

5.2 Comparison of computational performance

Finally, we study the performance of the B&B algorithm with growing datasets with the standard B&B algorithm as a benchmark. Only 4 models converge within the CPU time limit of 23h for at least one of the approaches including all 3 models with exact data, namely EOS262, EOS2262, and TSP, compare Table 4. For models EOS262 and EOS2262, the deterministic approach is the fastest with decreasing the runtime of the standard B&B algorithm by a factor of 3 and up to 4, respectively. For the TSP model, the MSE heuristics finds the global solution in the root node, resulting in a runtime of 2s, while the standard B&B algorithm takes about 2h for convergence with (SSE) and hits the CPU time limit with (MSE). Note that for exact data, the final lower bound equals the natural lower bound of 0 except for numerical tolerances. Thus, using reduced datasets allows for reducing the computational costs while retaining the tightness of the bounds on the optimal solution.

Contrarily, the standard B&B algorithm performs best for the EIS model: it converges within 8h, whereas all approaches of the B&B algorithm with growing datasets hit the CPU time limit. In the EIS model, we fit a model with 7 unknown parameters to a full dataset $\mathcal{D} \subsetneq \mathbb{R} \times \mathbb{R}^2$ with $|\mathcal{D}| = 26$ data points and, with the default settings, to reduced datasets containing 3, 10, 17, and 24 data points. Consequently, this parameter estimation problem is prone to overfitting, in particular, when using reduced datasets. In other words, the computational savings due to the data reduction come at the cost of looser lower bounds. Out-of-sample estimation allows for improving upon the quality of the lower bounds calculated based on reduced datasets, cf. Figure 6.

Figure 6 provides an overview of the range of final lower and upper bounds obtained by the different algorithms within the CPU time limit. Note that the lower bounds include, where applicable, the correction performed in the post-processing step and that the exact values are given in Online Resource 1. The first row of Figure 6 contains the models considering exact data such that the final lower bound is equal to zero except for numerical tolerances and the optimal solution is in the range of the optimality tolerance. For the remaining models, all approaches give a similar or even the same solution after the CPU time limit, compare the upper bounds depicted in Figure 6. For 7 out of 10 models considering noisy data, the SSE heuristic gives the best, i.e., largest, final lower bound. For 2 of these 10 models, the SSE heuristic and the standard B&B algorithm perform similarly. Only for IHMcon, the standard B&B algorithm finds the best lower bound. In accordance with the results of Section 5.1, we therefore conclude that the SSE heuristics allows to significantly increase the final lower bound within a given CPU time limit.

Table 4 Total CPU time needed for convergence of the B&B algorithm. Only models which converge within the CPU time limit for at least one of the algorithmic approaches, namely the standard B&B algorithm with objective (SSE) and (MSE), or the deterministic and heuristic approaches of the B&B algorithm with growing datasets using different augmentation rules are listed. Bold numbers indicate the fastest runtime for the respective model

	Standard B&B		Deterministic Appr. (SSE)			SSE heuristic			MSE heuristic		
Model	(SSE)	(MSE)	CONST	SCALING	COMBI	CONST	SCALING	TOL	CONST	OOS	TOL
Without resampling											
EOS262	2.0h	2.1h	1.7h	44min	**40** min	13.3h	–[a]	–[a]	7.9h	–[a]	5.8h
EOS2262	–[a]	–[a]	–[a]	**7.5** h	**7.5** h	–[a]	–[a]	–[a]	–[a]	–[a]	–[a]
EIS	9.2h	**8.0** h	–[a]	–[a]	–[a]	–[a]	–[a]	–[a]	–[a]	–[a]	–[a]
TSP	1.9h	–[a]	–[a]	40min	–[a]	–[a]	–[a]	9.2h	**2** s	**2** s	**2** s
With resampling											
EOS262	2.0h	2.1h	1.8h	47min	**38** min	12.8h	–[a]	–[a]	12.1h	–[a]	5.7h
EOS2262	–[a]	–[a]	–[a]	**5.5** h	6.0h	–[a]	–[a]	–[a]	–[a]	–[a]	–[a]
EIS	9.2h	**8.0** h	–[a]	–[a]	–[a]	–[a]	–[a]	–[a]	–[a]	–[a]	–[a]
TSP	1.9h	–[a]	–[a]	1.0h	–[a]	–[a]	–[a]	7.1h	**2** s	**2** s	**2** s

[a] Hitting CPU time limit of 23h

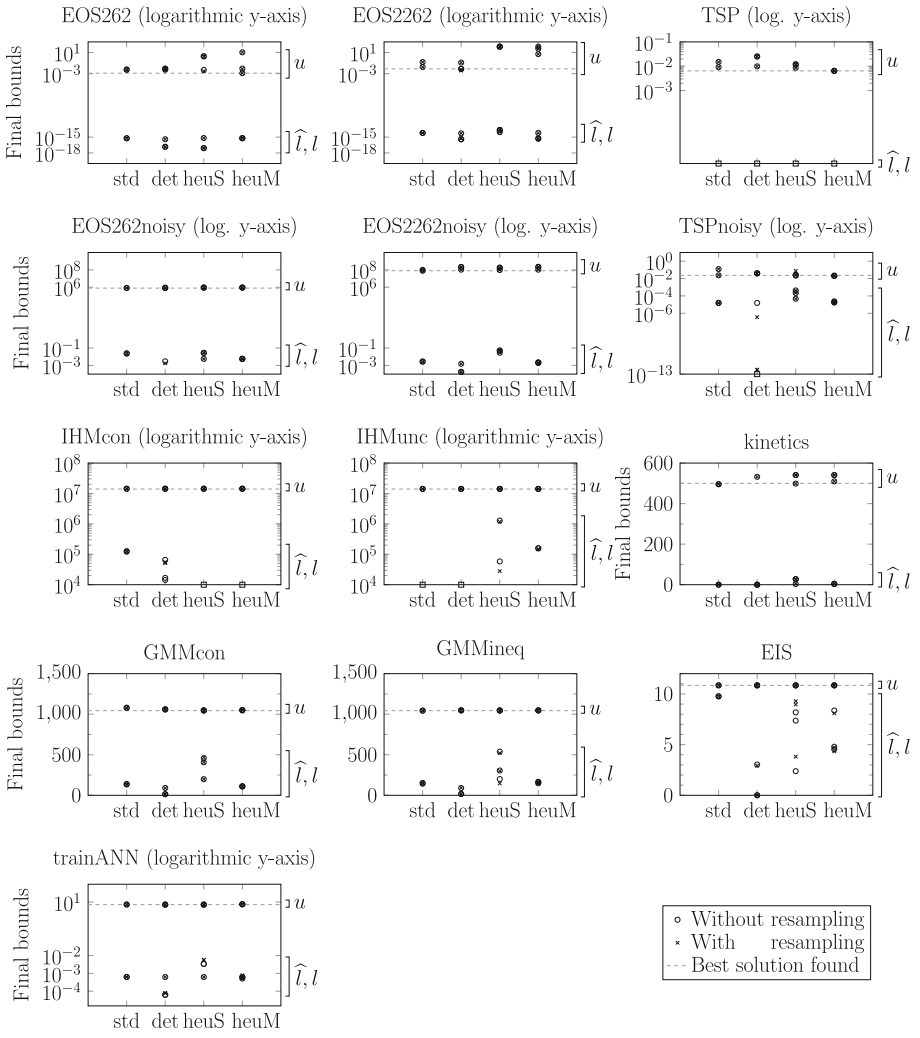

Fig. 6 Final lower bounds l_k or $\widehat{l}_k$ (strictly below dashed line) and final upper bounds u (above dashed line) on the optimal solution of the models for any of the approaches, namely the standard B&B algorithm (std) with objectives (SSE) and (MSE) as well as the deterministic approach (det), the SSE heuristic (heuS) and the MSE heuristic (heuM) of the B&B algorithm with growing datasets using the augmentation rules depicted in Figures 3 and 4. If applicable, the final lower bounds are subject to post-processing. The squares in the logarithmic plots indicate lower bounds equal to 0

We recall that models IHMunc and IHMcon are mathematically equivalent. While the optimization variables are coupled via equality constraints in the former, we use this relation as an explicit evaluation function for the latter, compare also the full-space and reduced-space formulations discussed in [2, 54]. While the final lower bound reported for IHMunc differs from 0 only when using the heuristic approaches, only the standard B&B algorithm and the deterministic approach determine lower bounds larger than 0 for IHMcon. The best lower bound for model IHMunc is in order of magnitude 10^6, while the best lower bound for model IHMcon is in order of magnitude 10^5. Thus, applying the heuristic approaches of the

B&B algorithm with growing datasets to the reduced-space formulation performs best for the IHM model. In analogy, the equality constraint in GMMcon is used to fix an unknown parameter in GMMineq. In this case, the reduced-space formulation GMMineq seems to be slightly advantageous for finding a good upper bound, while the lower bounds obtained for models GMMcon and GMMineq are similar for the different approaches.

When comparing the CPU times with and without resampling, cf. Table 4, there is no general tendency. On the one hand, resampling requires more computational resources. On the other hand, resampling affects the lower bound. Since MAiNGO uses the solution point of the lower bounding problem to initialize the local solver for the upper bounding problem, we may obtain a much better upper bound with resampling by chance. In this case, we need significantly less iterations for convergence. Apart from this, resampling does not seem to affect the final lower bounds in most of the cases, see Figure 6. In fact, the deviation in the number of nodes processed over the 5 repetitive runs affects the results in some cases as much as the choice whether to use resampling, e.g., the final lower bounds reported for model EOS2262. Noting that the resampling heuristic was introduced to reduce the variances of the reduced lower bounds, we infer that the B&B algorithm with growing datasets does not suffer from large variances.

6 Conclusions and outlook

We investigate out-of-sample estimation for enhancing the B&B algorithm with growing datasets proposed in our previous work [10]. In detail, we combine the lower bound calculated in a B&B node based on a reduced dataset with an out-of-sample evaluation obtaining the *combined lower bound*. Although the combined lower bound is only a heuristic lower bound, we expect it to be close to the lower bound calculated based on the full dataset. We use this heuristic lower bound for extending the deterministic approach presented in [10] as well as for pruning which results in two heuristic approaches, the so-called SSE and MSE heuristic. To detect and quantify potential mistakes made by the SSE and MSE heuristic, we introduce a post-processing check of the final lower bound after the B&B algorithm terminated.

We compared the performance of the different approaches of the B&B algorithm with growing datasets and the standard B&B algorithm based on 13 real-world applications from both literature and our previous works. For the estimation problems with exact data, the deterministic approach of the B&B algorithm with growing datasets yields the fastest runtimes. Most of our problems with error-prone data do not converge within the given CPU time limit. For these models, all approaches, including the standard B&B algorithm, find similar upper bounds. The SSE heuristics yields the best, i.e., largest, final lower bound for 70% of the models with noisy data and a similar value as the standard B&B algorithm for 20% of these models. In turn, the SSE heuristic may introduce an error into the pruning procedure. However, our results suggest that both the SSE and the MSE heuristic are almost deterministically converging towards the global solution, where the proposed post-processing procedure allows to detect and quantify the exceptions from this assumption.

Apart from this, we show in theory that we can likely decrease the variance of the lower bounds calculated based on reduced datasets when resampling the dataset. However, the actual numerical performance of the B&B algorithm with growing datasets for the models of our case study is hardly affected by the use of the resampling heuristic. Since we always fit a comparatively small number of unknown parameters to a large dataset, we seem to obtain small variances for the reduced lower bounds making resampling techniques like bagging less

advantageous, cf. [20] and [15, Section 8.2]. Note that the good generalization performance of our solution of model trainANN, see Online Resource 1, is another indication for having small variances.

The computational advantage of the B&B algorithm with growing datasets depends on the choice of the reduced datasets. On the one hand, estimation results based on a reduced dataset may be distorted by overfitting or even identifiability issues if the full dataset is already small compared to the number of unknown parameters. We have shown that applying out-of-sample estimation allows for tightening the lower bound calculated based on the reduced dataset. However, if the data reduction is insignificant in absolute numbers, the resulting computational savings cannot compensate for the remaining gap to the full lower bound. As a consequence, the B&B algorithm with growing datasets is still outperformed by the standard algorithm in such cases with small datasets. On the other hand, we expect that the performance of the B&B algorithm with growing datasets profits from exploiting knowledge about the dataset for the data reduction. For example, the reduced lower bound may be a better fit to the full lower bound if the reduced datasets contain a sufficient amount of measurements with the largest measurement errors. An extension allowing for user-given reduced datasets merits therefore careful attention.

We make the parameter estimation problems of our case study openly accessible via repository GloPSE[2]. In future work, we aim at extending the case study with estimation problems containing data-dependent constraints, e.g., fitting binary fluid systems with constraints on the measured mole fractions [55] and the optimization of Gaussian processes with constraints on each of the data points of the training set [56]. For this, we aim at implementing a specific treatment of data-dependent constraints to allow for an efficient handling of reduced datasets for these models.

For all computations, we used our open-source solver MAiNGO which uses McCormick relaxations [43, 44] to calculate convex underestimators for the lower bounding problem. MAiNGO can therefore efficiently handle reduced-space formulations [2, 54], meaning that the size of the dataset does not affect the number of optimization variables but solely the computational effort for function evaluations. In contrast to that, many DGO solvers like BARON [57, 58], ANTIGONE [59], and SCIP [60, 61] use the auxiliary values method (AVM) [6, 62–64] for obtaining convex underestimators. An implementation of the B&B algorihm with growing datasets in these AVM-based solvers is of high interest since the AVM method may add an auxiliary optimization variable for each of the data points and we therefore expect AVM-based solvers to profit even stronger from the use of growing datasets.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10898-025-01514-4.

Acknowledgements This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant MI 1851/10-1 "Parameter estimation with (almost) deterministic global optimization". Simulations were performed with computing resources granted by RWTH Aachen University under project rwth1563. Susanne Sass is grateful for her association to the International Research Training Group (DFG) IRTG-2379 "Hierarchical and Hybrid Approaches in Modern Inverse Problems" under grant 333849990/GRK2379. We thank Dominik Bongartz for his support in the implementation of the novel approaches within MAiNGO. Moreover, we thank Ian H. Bell, Tanvir Rahat, Alexander Echtermeyer, and J. Raphael Seidenberg for their assistance with setting up the EOS, TSP, IHM, and EIS model, respectively. We are grateful to Ian H. Bell for providing the data for the EOS model as well as to Niklas Thissen, Stefanie Khan, and Anna K. Mechler for providing experimental measurements for the EIS model. Moreover, we thank the anonymous reviewer whose comments helped us with improving and clarifying this article.

Author Contributions Conceptualization: Susanne Sass, Angelos Tsoukalas; Methodology: Susanne Sass, Alexander Mitsos, Nikolay I. Nikolov (particularly of Section 4), Angelos Tsoukalas; Software: Susanne Sass;

Formal analysis and investigation: Susanne Sass; Writing - original draft preparation: Susanne Sass; Writing - review and editing: Susanne Sass, Alexander Mitsos, Nikolay I. Nikolov, Angelos Tsoukalas; Funding acquisition: Susanne Sass, Alexander Mitsos, Angelos Tsoukalas; Resources: Alexander Mitsos; Supervision: Alexander Mitsos.

Data Availability An implementation of the discussed algorithmic approaches within open-source solver MAiNGO is available at https://git.rwth-aachen.de/avt-svt/public/maingo. The exact datasets and implementations of the discussed models are published in open-source repository GloPSE at https://git.rwth-aachen.de/avt-svt/public/glopse.

Declarations

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

References

1. Singer, A.B., Taylor, J.W., Barton, P.I., Green, W.H.: Global Dynamic Optimization for Parameter Estimation In Chemical Kinetics. J. Phys. Chem. A **110**(3), 971–976 (2006). https://doi.org/10.1021/jp0548873
2. Mitsos, A., Chachuat, B., Barton, P.I.: McCormick-Based Relaxations of Algorithms. SIAM J. Optim. **20**(2), 573–601 (2009). https://doi.org/10.1137/080717341
3. Horst, R., Pardalos, P.M.: Handbook of Global Optimization vol. 2. Springer, New York, NY (1995). https://doi.org/10.1007/978-1-4615-2025-2
4. Floudas, C.A.: Deterministic Global Optimization: Theory, Methods and Applications vol. 37. Springer, New York, NY (2000). https://doi.org/10.1007/978-1-4757-4949-6
5. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches, 3rd edn. Springer, Berlin, Heidelberg (1996). https://doi.org/10.1007/978-3-662-03199-5
6. Tawarmalani, M., Sahinidis, N.V.: Convexification and Global Optimization in Continuous and Mixed-integer Nonlinear Programming: Theory, Algorithms, Software, and Applications vol. 65. Springer, New York, NY (2002). https://doi.org/10.1007/978-1-4757-3532-1
7. Bagirov, A.M., Ugon, J., Mirzayeva, H.: Nonsmooth nonconvex optimization approach to clusterwise linear regression problems. Eur. J. Oper. Res. **229**(1), 132–142 (2013). https://doi.org/10.1016/j.ejor.2013.02.059
8. Kovačević, D., Mladenović, N., Petrović, B., Milošević, P.: DE-VNS: Self-adaptive Differential Evolution with crossover neighborhood search for continuous global optimization. Comput. Oper. Res. **52**, 157–169 (2014). https://doi.org/10.1016/j.cor.2013.12.009
9. Kristianto, Y., Gunasekaran, A.: A global optimization for sustainable multi-domain global manufacturing. Comput. Oper. Res. **89**, 307–323 (2018). https://doi.org/10.1016/j.cor.2015.12.001
10. Sass, S., Mitsos, A., Bongartz, D., Bell, I.H., Nikolov, N.I., Tsoukalas, A.: A branch-and-bound algorithm with growing datasets for large-scale parameter estimation. Eur. J. Oper. Res. **316**(1), 36–45 (2024). https://doi.org/10.1016/j.ejor.2024.02.020
11. Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.-T.: Learning from Data: A Short Course. AMLBook, New York (2012)
12. Murphy, K.P.: Probabilistic Mach. Learn.: An Introduction. Adaptive computation and Mach. Learn. The MIT press, Cambridge, Massachusetts (2022)
13. Quenouille, M.H.: Notes on Bias in Estimation. Biometrika **43**(3/4), 353 (1956). https://doi.org/10.2307/2332914

14. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Monographs on statistics and applied probability. Chapman & Hall/CRC, New York (1994). https://doi.org/10.1201/9780429246593
15. James, G., Hastie, T., Witten, D., Tibshirani, R.: An Introduction to Statistical Learning: with Applications in R. Springer, New York, NY (2013). https://doi.org/10.1007/978-1-4614-7138-7
16. Bongartz, D., Najman, J., Sass, S., Mitsos, A.: MAiNGO – McCormick-based Algorithm for mixed-integer Nonlinear Global Optimization. http://permalink.avt.rwth-aachen.de/?id=729717 [Accessed Apr 17, 24] (2018)
17. Locatelli, M., Schoen, F.: Global Optimization: Theory, Algorithms, and Applications vol. 15. MOS-SIAM Ser. Optim., Philadelphia, PA (2013). https://doi.org/10.1137/1.9781611972672
18. Hastie, T.J., Friedman, J.H., Tibshirani, R.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, NY (2017). https://doi.org/10.1007/978-0-387-84858-7
19. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
20. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996). https://doi.org/10.1007/BF00058655
21. Lemmon, E.W., McLinden, M.O., Wagner, W.: Thermodynamic Properties of Propane. III. A Reference Equation of State for Temperatures from the Melting Line to 650 K and Pressures up to 1000 MPa. J. Chem. Eng. Data **54**(12), 3141–3180 (2009) https://doi.org/10.1021/je900217v
22. Sass, S., Tsoukalas, A., Bell, I.H., Bongartz, D., Najman, J., Mitsos, A.: Towards global parameter estimation exploiting reduced data sets. Optim. Methods Softw. **38**(6), 1129–1141 (2023). https://doi.org/10.1080/10556788.2023.2205645
23. Alsmeyer, F., Koss, H.-J., Marquardt, W.: Indirect spectral hard modeling for the analysis of reactive and interacting mixtures. Appl. Spectrosc. **58**(8), 975–985 (2004). https://doi.org/10.1366/0003702041655368
24. Echtermeyer, A., Marks, C., Mitsos, A., Viell, J.: Inline Raman Spectroscopy and Indirect Hard Modeling for Concentration Monitoring of Dissociated Acid Species. Appl. Spectrosc. **75**(5), 506–519 (2021). https://doi.org/10.1177/0003702820973275
25. Echtermeyer, A.W.W., Marks, C., Mitsos, A., Viell, J.: Dataset to "Inline Raman Spectroscopy and Indirect Hard Modeling for Concentration Monitoring of Dissociated Acid Species", RWTH Aachen University (2024). https://doi.org/10.18154/RWTH-2024-01177
26. Taylor, J.W., Ehlker, G., Carstensen, H.-H., Ruslen, L., Field, R.W., Green, W.H.: Direct Measurement of the Fast, Reversible Addition of Oxygen to Cyclohexadienyl Radicals in Nonpolar Solvents. The J. Phys. Chem. A **108**(35), 7193–7203 (2004). https://doi.org/10.1021/jp0379547
27. Sass, S., Seidenberg, J.R.: GloPSE/ElectrochemicalImpedanceSpectroscopy2CPE. https://git.rwth-aachen.de/avt-svt/public/glopse [Accessed Apr 23, 2024] (2024)
28. Thissen, N., Hoffmann, J., Tigges, S., Vogel, D.A.M., Thoede, J.J., Khan, S., Schmitt, N., Heumann, S., Etzold, B.J.M., Mechler, A.K.: Industrially Relevant Conditions in Lab-Scale Analysis for Alkaline Water Electrolysis. ChemElectroChem **11**(1), 202300432 (2024). https://doi.org/10.1002/celc.202300432
29. Thissen, N., Khan, S., Mechler, A.K.: Industrially relevant characterisation of a Ni mesh anode in alkaline water electrolysis (2024). https://doi.org/10.5281/zenodo.11103701
30. Villaverde, A.F., Fröhlich, F., Weindl, D., Hasenauer, J., Banga, J.R.: Benchmarking optimization methods for parameter estimation in large kinetic models. Bioinformatics **35**(5), 830–838 (2019). https://doi.org/10.1093/bioinformatics/bty736
31. Moles, C.G., Mendes, P., Banga, J.R.: Parameter estimation in biochemical pathways: a comparison of global optimization methods. Genome Res. **13**(11), 2467–2474 (2003). https://doi.org/10.1101/gr.1262503
32. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. J. R. Stat. Soc. Ser. B. Stat. Methodol. **39**(1), 1–38 (1977). https://doi.org/10.1111/j.2517-6161.1977.tb01600.x
33. Härdle, W.: Smoothing Techniques: With Implementation in S. Springer series in statistics. Springer, New York, NY (1991). https://doi.org/10.1007/978-1-4612-4432-5
34. Azzalini, A., Bowman, A.W.: A Look at Some Data on the Old Faithful Geyser. Appl. Stat. **39**(3), 357–365 (1990). https://doi.org/10.2307/2347385
35. Hinton, G.E.: Learning translation invariant recognition in a massively parallel networks. In: Goos, G., Hartmanis, J., Barstow, D., Brauer, W., Brinch Hansen, P., Gries, D., Luckham, D., Moler, C., Pnueli, A., Seegmüller, G., Stoer, J., Wirth, N., Bakker, J.W., Nijman, A.J., Treleaven, P.C. (eds.) PARLE Parallel Architectures and Languages Europe. Lecture Notes in Computer Science, vol. 258, pp. 1–13. Springer, Berlin, Heidelberg (1987). https://doi.org/10.1007/3-540-17943-7_117

36. Schweidtmann, A.M., Mitsos, A.: Deterministic Global Optimization with Artificial Neural Networks Embedded. J. Optim. Theory Appl. **180**(3), 925–948 (2019). https://doi.org/10.1007/s10957-018-1396-0
37. Fanaee-T, H.: Bike Sharing Dataset. UCI Mach. Learn. Repository (2013). https://doi.org/10.24432/C5W894
38. Fanaee-T, H., Gama, J.: Event labeling combining ensemble detectors and background knowledge. Prog. Artif. Intell. **2**(2–3), 113–127 (2014). https://doi.org/10.1007/s13748-013-0040-3
39. Bussieck, M.R., Drud, A.S., Meeraus, A.: MINLPLib–A Collection of Test Models for Mixed-Integer Nonlinear Programming. INFORMS J. Comput. **15**(1), 114–119 (2003). https://doi.org/10.1287/ijoc.15.1.114.15159
40. Vanderbei, R., colleagues: Nonlinear Optimization Models, note = https://vanderbei.princeton.edu/ampl/nlmod [Accessed Feb 26, 2024] (2004)
41. Shcherbina, O., Neumaier, A., Sam-Haroud, D., Vu, X.-H., Nguyen, T.-V.: Benchmarking Global Optimization and Constraint Satisfaction Codes. In: Bliek, C., Jermann, C., Neumaier, A. (eds.) Global Optimization and Constraint Satisfaction, pp. 211–222. Springer, Berlin, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39901-8_16
42. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006). https://doi.org/10.1007/s10107-004-0559-y
43. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I - Convex underestimating problems. Math. Program. **10**(1), 147–175 (1976). https://doi.org/10.1007/BF01580665
44. Tsoukalas, A., Mitsos, A.: Multivariate McCormick relaxations. J. Glob. Optim. **59**(2), 633–662 (2014). https://doi.org/10.1007/s10898-014-0176-0
45. Chachuat, B., Houska, B., Paulen, R., Perić, N., Rajyaguru, J., Villanueva, M.E.: Set-Theoretic Approaches in Analysis, Estimation and Control of Nonlinear Systems. IFAC-PapersOnLine **48**(8), 981–995 (2015) https://doi.org/10.1016/j.ifacol.2015.09.097 https://github.com/omega-icl/mcpp [Retrieved Nov 11, 2019]
46. Forrest, J.J., Vigerske, S., Ralphs, T., Hafer, L., Fasano, J.P., Santos, H.G., Saltzman, M., Gassmann, H., Kristjansson, B., King, A.: COIN-OR Linear Programming Solver (2019). https://github.com/coin-or/Clp [Retrieved Nov 8, 2019]
47. International Business Machines Corporation: IBM ILOG CPLEX Optimization Studio v22.1.1, Armonk, NY (2022)
48. Kulisch, U.: C++ Toolbox for Verified Computing I: Basic Numerical Problems Theory, Algorithms, and Programs. Springer eBook Collection Mathematics and Statistics. Springer, Berlin, Heidelberg (1995). https://doi.org/10.1007/978-3-642-79651-7
49. Lerch, M., Tischler, G., Gudenberg, J.W., Hofschuster, W., Krämer, W.: FILIB++, a fast interval library supporting containment computations. ACM Trans. Math. Softw. **32**(2), 299–324 (2006). https://doi.org/10.1145/1141885.1141893
50. Gleixner, A.M., Berthold, T., Müller, B., Weltge, S.: Three enhancements for optimization-based bound tightening. J. Glob. Optim. **67**(4), 731–757 (2017). https://doi.org/10.1007/s10898-016-0450-4
51. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**(1–3), 503–528 (1989). https://doi.org/10.1007/BF01589116
52. Nocedal, J.: Updating Quasi-Newton Matrices with Limited Storage. Math. Comput. **35**(151), 773–782 (1980). https://doi.org/10.2307/2006193
53. Johnson, S.G.: The NLopt nonlinear-optimization package. http://github.com/stevengj/nlopt [Retrieved Nov 6, 2019] (2024)
54. Bongartz, D., Mitsos, A.: Deterministic global optimization of process flowsheets in a reduced space using McCormick relaxations. J. Glob. Optim. **69**(4), 761–796 (2017). https://doi.org/10.1007/s10898-017-0547-4
55. Bollas, G.M., Barton, P.I., Mitsos, A.: Bilevel Optimization Formulation for Parameter Estimation in Vapor-Liquid(-Liquid) Phase Equilibrium Problems. Chem. Eng. Sci. **64**(8), 1768–1783 (2009). https://doi.org/10.1016/j.ces.2009.01.003
56. Schweidtmann, A.M., Bongartz, D., Grothe, D., Kerkenhoff, T., Lin, X., Najman, J., Mitsos, A.: Deterministic global optimization with Gaussian processes embedded. Math. Program. Comput. **13**(3), 553–581 (2021). https://doi.org/10.1007/s12532-021-00204-y
57. Tawarmalani, M., Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. Math. Program. **103**(2), 225–249 (2005). https://doi.org/10.1007/s10107-005-0581-8
58. Khajavirad, A., Sahinidis, N.V.: A hybrid LP/NLP paradigm for global optimization relaxations. Math. Program. Comput. **10**(3), 383–421 (2018). https://doi.org/10.1007/s12532-018-0138-5

59. Misener, R., Floudas, C.: ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. J. Glob. Optim. **59**, 503–526 (2014). https://doi.org/10.1007/s10898-014-0166-2
60. Achterberg, T.: SCIP: solving constraint integer programs. Math. Program. Comput. **1**(1), 1–41 (2009). https://doi.org/10.1007/s12532-008-0001-1
61. Vigerske, S., Gleixner, A.: SCIP: Global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. Optim. Methods Softw. **33**(3), 563–593 (2018). https://doi.org/10.1080/10556788.2017.1335312
62. Smith, E.M.B., Pantelides, C.C.: Global optimisation of nonconvex MINLPs. Comput. Chem. Eng. **21**, 791–796 (1997). https://doi.org/10.1016/S0098-1354(97)87599-0
63. Smith, E.M.B., Pantelides, C.C.: A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. Comput. Chem. Eng. **23**(4–5), 457–478 (1999). https://doi.org/10.1016/S0098-1354(98)00286-5
64. Ryoo, H.S., Sahinidis, N.V.: A branch-and-reduce approach to global optimization. J. Glob. Optim. **8**(2), 107–138 (1996). https://doi.org/10.1007/BF00138689