Global Parameter Estimation for the Efficient Optimization of Real-World Applications

Globale Parameterschätzung für die effiziente Optimierung von realen Anwendungsproblemen

Von der Fakultät für Maschinenwesen der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades einer Doktorin der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Susanne Sass geb. Scholl

Berichter: Universitätsprofessor Alexander Mitsos, Ph.D.
Assistant Professor Angelos Tsoukalas, Ph. D.

Tag der mündlichen Prüfung: 29.11.2024

Titel: Global Parameter Estimation for the Efficient Optimization of Real-World Applications

Autor: Susanne Sass

Reihe: Aachener Verfahrenstechnik Series
AVT.SVT - Process Systems Engineering
Band 34 (2024)

Herausgeber: Aachener Verfahrenstechnik
Forckenbeckstraße 51
52074 Aachen
Tel.: +49 (0)241 80 97717
Fax.: +49 (0)241 80 92326
E-Mail: secretary.svt@avt.rwth-aachen.de
https://www.avt.rwth-aachen.de

Volltext verfügbar: 10.18154/RWTH-2024-11336

Preface

I am deeply grateful for the opportunity to work, grow, and doing my doctorate at the chair of Process Systems Engineering (AVT.SVT) at RWTH Aachen University. First and foremost, I would like to thank my advisor Alexander Mitsos for his support, his open-mindedness, his straightforwardness, and all the easygoing or highly complex discussions we have had. Special thanks also to my second examiner Angelos Tsoukalas for all the valuable feedback and fruitful discussions. Moreover, I wish to gratefully acknowledge the chairmanship of Andrea Benigni at my doctoral examination.
Besides, I would like to thank Timm Faulwasser, Dinah Hollermann, Dominique Sauer, Veit Hagenmeyer, and all the other members of ES2050 for the great collaboration within the project. Moreover, I owe special thanks to Ian Bell, Nikolay Nikolov, and Sebastian Sager. Many improvements of my thesis and publications are the result of their comments and suggestions.
Thanks are also due to Adel Mhamdi, Sven Groß, Preet Joy, and Georgianna Prokopou for the enjoyable collaboration within my teaching duties. A sincere note of appreciation goes to Hatim Djelassi, Jaromił Najman, Ralf Hannemann-Tamás, Dominik Bongartz, Moll Glass, Chryssa Kappatou, Marco Langiu, Aron Zingler, Clara Witte, Chrysanthi Papadimitriou, and many, many other colleagues. All of you created an imaginative and supportive working environment which was essential for the success of my doctorate. I would also like to thank Wanda Frohn, Petra Eissa, Didem Uslu and her team as well as Jutta Friedrich, Sascha Gerhards, and their team for their continued support such that I could focus on the core of my job.
Last but not least I would like to express my heartfelt gratitude to Sabine Hölzer-Pöll and my family, particularly to my husband Hauke. Things would not be the same without you.

Hamburg, December 2024 *Susanne Sass*

Contents

Kurzfassung

Mathematische Optimierung ermöglicht es, Kenntnisse über industrielle, thermodynamische, biochemische und viele andere Prozesse zu gewinnen und auszunutzen. Aufgrund der Komplexität technischer und natürlicher Prozesse sind realitätsnahe Optimierungsprobleme jedoch nur lösbar, wenn sowohl eine effiziente mathematische Problemformulierung als auch ein effizienter Optimierungsalgorithmus verwendet wird. Daher untersuchen wir einerseits die Grenzen einer typischen Annahme zur Verringerung der Modellkomplexität und entwickeln andererseits einen neuartigen Optimierungsalgorithmus.
Zuerst untersuchen wir die Annahme eines quasi-stationären Betriebs, die häufig der Optimierung der Auslegung von Energieversorgungsanlagen zugrunde liegt. Obwohl quasi-stationäre Modelle die Optimierung von größeren, also auch von realitätsnäheren, Energiesystemen ermöglichen, sind die Ergebnisse nur realistisch, wenn das dynamische Verhalten ausreichend genau abgebildet wird. Im ersten Schritt identifizieren wir die Modellgleichungen als lineare Differentialgleichungen 1. Ordnung, bei denen die inhärente Trägheit über eine Zeitkonstante charakterisiert werden kann. Unter Berücksichtigung schwankender Energiebedarfe und Strompreise untersuchen wir die Auswirkung verschiedener Trägheiten auf den optimalen Betrieb für einen Heizungskessel im Detail sowie für ein komplexes Energiesystem. Die numerischen Ergebnisse deuten darauf hin, dass bei der Verwendung von quasi-stationären Modellen Rampenbedingungen notwendig sind. Selbst unter Verwendung dynamischer Modelle mit geringen Trägheiten treten Veränderungen in der optimalen Betriebsführung auf. Ein Wärmespeicher kann bis zu einem gewissen Grad die Trägheiten des Systems kompensieren, das dynamische Verhalten jedoch nicht in einen quasi-stationären Betrieb überführen.
Im zweiten Teil der Arbeit erweitern wir die Grenzen deterministischer globaler Optimierung, indem wir die allgemeine Struktur von Parameterschätzproblemen ausnutzen. Genauer gesagt vereinfachen wir das Schätzproblem zu Beginn eines Branch-and-Bound-Algorithmus (B&B), indem wir einen reduzierten Datensatz betrachten. Im Fortlaufen des Algorithmus ergänzen wir Datenpunkte, sodass der Datensatz auf seine ursprüngliche Größe "wächst". Wir beweisen, dass der *B&B-Algorithmus mit wachsenden Datensätzen* gegen die global optimale Lösung konvergiert. Da das simple Weglassen von Datenpunkten das Konvergenzverhalten des B&B-Algorithmus negativ beeinflussen kann, leiten wir zudem unter Verwendung sogenannter Out-of-sample-Fehler zwei heuristische Ansätze ab. Unsere Rechenstudie lässt den Schluss zu, dass wir potenzielle Fehler der heuristischen Methoden mittels eines Post-Processing-Schrittes detektieren und quantifizieren können. Für den überwiegenden Teil der untersuchten realen Schätzprobleme liefert der neu entwickelte Algorithmus eine bessere Rechenleistung als der Standardalgorithmus. Abschließend folgern wir aus Vergleichsergebnissen mit einer Resampling-Heuristik, dass die Verwendung von zufällig gewählten Datenpunkten keine großen Varianzen verursacht.
Die Kombination des B&B-Algorithmus mit wachsenden Datensätzen mit dynamischer Optimierung hat großes Potential, die Optimierung weiterer herausfordernder Anwendungsprobleme mittels effizienter Algorithmen zu ermöglichen.

Summary

Numerical optimization allows for gaining and exploiting insights into industrial, thermodynamic, biochemical, and various other processes. However, due to the complexity of technical and natural processes, the optimization of the related real-world applications is only tractable for existing optimization solvers when using both an efficient model formulation and optimization algorithm. Thus, we investigate the limitations of a common assumption for reducing the model's complexity on the one hand and propose a novel optimization approach on the other hand.

First, we focus on steady-state assumptions commonly used in the design optimization of energy supply systems for the underlying operation. Although quasi-steady models allow for the optimization of larger and, thus, closer-to-reality energy supply systems, the optimal results are only realistic if the dynamic operation of the system components is adequately represented. To that end, we identify the dynamic behavior of the system components as first-order dynamics where the inherent inertia is characterized by a time constant. Considering volatile energy demands and electricity prices, we study the impact of different time constants on the optimal operation of a boiler in more detail and perform a numerical study with a multi-component energy supply system. Our results suggest that ramping constraints are required when using quasi-steady models for modeling the dynamic operation of energy supply systems since even fast dynamics result in changes of the optimal operational strategy. A thermal energy storage may compensate slow dynamics to some extent but cannot transform dynamic behavior into quasi-steady behavior.

Second, we enhance the capabilities of deterministic global optimization methods by exploiting the general structure of parameter estimation problems. In fact, we simplify the estimation problem at the beginning of the spatial B&B algorithm by reducing the dataset. When advancing with the algorithm, we let the dataset grow to its original size by augmenting data points. We prove that the *B&B algorithm with growing datasets* is guaranteed to converge towards the global optimal solution. Since simply omitting data points may decrease the benefits of the B&B algorithm with growing datasets, we additionally propose two heuristic approaches utilizing out-of-sample estimation. Our benchmark study suggests that we can detect and quantify potential mistakes made by the heuristic approaches based on a post-processing procedure. For the vast majority of the investigated real-world estimation problems, the B&B algorithm with growing datasets shows a superior computational performance compared to the standard B&B algorithm. Finally, the application of a resampling heuristic indicates that randomly drawing data points for the initial dataset does not lead to large variances.

The combination of both dynamic optimization and the B&B algorithm with growing datasets is an intriguing future perspective for investigating further challenging real-world applications with the help of efficient optimization algorithms.

Publications and Copyrights

This thesis is based on the research performed by the author during her time at the Chair of Process Systems Engineering (AVT.SVT) at RWTH Aachen University between September 2016 and July 2024. Parts of this thesis have already been published or submitted for publication as described in the following. Only parts of the publications representing the author's work are used in this thesis.

- The majority of Chapter 2, excluding the case study on energy system ES1, and of Appendix A.2 is reproduced in part from S. Sass and A. Mitsos. Optimal Operation of Dynamic (Energy) Systems: When are Quasi-Steady Models Adequate? *Computers & Chemical Engineering*, 124:133–139, 2019. doi:10.1016/j.compchemeng.2019.02.011. Copyright © 2019 Elsevier Ltd.

- Part of the introduction in Section 2.1 as well as the case study on energy system ES1 discussed in Section 2.3.3 and Appendix A.1 are reproduced in part from S. Sass, T. Faulwasser, D. E. Hollermann, C. D. Kappatou, D. Sauer, T. Schütz, D. Y. Shu, A. Bardow, L. Gröll, V. Hagenmeyer, D. Müller, and A. Mitsos. Model compendium, data, and optimization benchmarks for sector-coupled energy systems. *Computers & Chemical Engineering*, 135:106760, 2020. doi:10.1016/j.compchemeng.2020.106760. Copyright © 2020 Elsevier Ltd.

- The description of MAiNGO's capabilities for mixed-integer nonlinear programs in Section 3.2.2 is adapted from the technical report D. Bongartz, J. Najman, S. Sass, and A. Mitsos. MAiNGO – McCormick-based Algorithm for mixed-integer Nonlinear Global Optimization. Process Systems Engineering (AVT.SVT), RWTH Aachen University, 2018. URL `http://permalink.avt.rwth-aachen.de/?id=729717`, Last accessed: Dec 1, 2024. Copyright © 2018 AVT.SVT

- Part of the description of the standard branch-and-bound algorithm in Section 3.2.2 and of the EOS model in Appendix D.3.2.1 as well as the case study on the deterministic global optimization of reduced models presented in Section 3.3 and Appendix B are reproduced in part from S. Sass, A. Tsoukalas, I. H. Bell, D. Bongartz, J. Najman, and A. Mitsos. Towards global parameter estimation exploiting reduced data sets. *Optimization Methods and Software*, 38(6):1129–1141, 2023. doi:10.1080/10556788.2023.2205645. Copyright © 2023 Taylor & Francis Group

- The literature review presented in Section 3.1.1, the introduction of the general parameter estimation problem in Section 3.2.1, the theoretical foundation and description of the branch-and-bound algorithm with growing datasets discussed in Section 3.4 and Appendix C excluding augmentation rules COMBI and TOL, the computational results for single branch-and-bound nodes presented in Section 3.6.1, the literature review on existing benchmark libraries in Section 3.6.2, the computational

results for models EOS262, EOS2262, and TSPexact discussed in Section 3.6.2.2, and the setup of model TSPexact in Appendix D.3.7.1 are reproduced in part from the main article and supplementary material of S. Sass, A. Mitsos, D. Bongartz, I. H. Bell, N. I. Nikolov, and A. Tsoukalas. A branch-and-bound algorithm with growing datasets for large-scale parameter estimation. *European Journal of Operational Research*, 316(1):36–45, 2024. doi:10.1016/j.ejor.2024.02.020. Copyright © 2024 Elsevier B.V.

– The introduction of augmentation rules COMBI and TOL in Sections 3.4.2.3 and 3.4.2.4, respectively, the most part of Section 3.5 about the heuristic pruning approaches and of Section 3.6.2 about the extensive case study benchmarking the branch-and-bound algorithm with growing datasets, the discussion of the resampling heuristic in Section 3.6.3, as well as the computational results and model formulations in Appendix D are reproduced in part from the main article and supplementary material of S. Sass, A. Mitsos, N. I. Nikolov, and A. Tsoukalas. Out-of-sample estimation for a branch-and-bound algorithm with growing datasets. *Submitted*, 2024. Only parts of the publication representing the author's work are used in this thesis.

Alexander Mitsos provided ideas and guidance as well as edits to all listed publications and this thesis. Angelos Tsoukalas provided ideas to the content of Chapter 3 and the related publications [174–176]. Student assistant Tanvir Rahat helped with preparing model TSP for the optimization with MAiNGO, cf. Section D.3.7.
Moreover, the author contributed to the following publications which do not form part of this thesis:

– M. N. Jung, C. Kirches, S. Sager, and S. Sass. Computational Approaches for Mixed Integer Optimal Control Problems with Indicator Constraints. *Vietnam Journal of Mathematics*, 46(4):1023–1051, 2018. doi:10.1007/s10013-018-0313-z

– A. Caspari, A. M. Bremen, J. M. Faust, F. Jung, C. D. Kappatou, S. Sass, Y. Vaupel, R. Hannemann-Tamás, A. Mhamdi, and A. Mitsos. DyOS - A Framework for Optimization of Large-Scale Differential Algebraic Equation Systems. In Anton A. Kiss, Edwin Zondervan, Richard Lakerveld, and Leyla Özkan, editors, *29th European Symposium on Computer Aided Process Engineering*, volume 46 of *Computer Aided Chemical Engineering*, pages 619–624. Elsevier, 2019. doi:10.1016/B978-0-12-818634-3.50104-1

– C. D. Kappatou, D. Bongartz, J. Najman, S. Sass, and A. Mitsos. Global dynamic optimization with Hammerstein-Wiener models embedded. *Journal of Global Optimization*, 84(2):321–347, 2022. doi:10.1007/s10898-022-01145-z

During her time at AVT.SVT, the author supervised or co-supervised the student theses of Thomas Hosten (2017), Aron Zingler (2018), and Romeo Valentin (2019) as well as the work of student assistant Prakhar Dutta. None of these works form part of this thesis.
The work for this thesis was funded by the Helmholtz Association under the Joint Initiative "Energy System 2050 – A Contribution of the Research Field Energy" and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant MI 1851/10-1 "Parameter estimation with (almost) deterministic global optimization". A part of the simulations were performed with computing resources granted by RWTH Aachen University

under project rwth1563. The author 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.

General notation

The following listings introduce abbreviations and uniform notation used in wide parts of this thesis. They do not form an exhaustive enumeration. Further symbols as well as specific subscripts and superscripts are declared close to their usage.

Abbreviations

AVM	Auxiliary variables method
B&B	Branch-and-bound
CHP unit	Combined heat and power unit
DGO	Deterministic global optimization
DO	Dynamic optimization problem
LBP	Lower bounding problem
LP	Linear program
MILP	Mixed-integer linear program
MINLP	Mixed-integer nonlinear program
MSE	Mean squared error
NLP	Nonlinear program
OCP	Optimal control problem
PE	Parameter estimation problem
PV unit	Photovoltaic unit
SSE	Summed squared error
TES	Thermal energy storage

Symbols

Mathematical operations

$\lvert \cdot \rvert$	Cardinality of a set
$\lvert\lvert \cdot \rvert\rvert$	Euclidean norm
$\lceil \cdot \rceil$	Ceiling function (Rounding up to nearest integer)
$\mathrm{rd}\,(\cdot)$	Rounding to nearest integer
$\mathrm{EV}\,[\cdot]$	Expected value of a random variable
$\mathbb{P}(\cdot)$	Probability of an event
$\mathbb{R}$	Real numbers
$\mathbb{Z}$	Integers

Efficient optimization of dynamic energy systems

η	Efficiency
τ	Time constant
E	Thermal energy
l	Part-load fraction
P	Electric power
$\dot{Q}$	Heat transfer rate
t	Time
T	Temperature

Global parameter estimation exploiting reduced datasets

ε	Optimality tolerance
$\mathcal{D}$	Dataset
k	Index of B&B iteration
l	Valid lower bound
$\widehat{l}$	Heuristic lower bound
N	B&B node
$\mathcal{N}$	Set of B&B nodes
$\boldsymbol{p}$	Unknown parameters (Optimization variables)
$\mathcal{P}$	Parameter domain
u	(Valid) Upper bound
$\boldsymbol{x}$	Input data
y	Output data

1. Introduction and overview

Mathematical programming allows to improve the economic and ecologic performance of industrial processes without the need for costly experiments, e.g., via the numerical optimization of design and operation of power plants and energy supply systems [26, 47, 71, 84, 85, 183]. However, numerical optimization can only improve real-world processes if the applied computational methods terminate with an optimal solution and the underlying mathematical models represent real-world data adequately. To that end, we solve parameter estimation problems for fitting the unknown model parameters based on a provided measurement dataset.

This thesis starts from the premise that the quantity and quality of the dataset provided with the mathematical model as well as the structure of the model equations allows for determining the unknown parameters. For related research beyond this premise, we refer to the approaches for optimization under uncertainty, see, e.g., [82, 113, 133, 156, 157], and optimal experimental design [e.g., 10, 83] as well as the concepts of structural identifiability [21, 105].

We aim at developing *efficient* optimization approaches by (*i*) using concise mathematical models, (*ii*) overcoming suboptimal solutions, and (*iii*) exploiting standard approaches from other fields of research. With (*i*), we refer to the simplest model which yields realistic results. Generally, this requires to challenge modeling assumptions like the level of detail, e.g., modeling spatial temperature distributions within energy conversion components versus modeling only the interchanged energy flow rates; the choice of the time scale, e.g., modeling based on seasons, hours or seconds; or steady-state assumptions. In this thesis, we illustrate the applicability of dynamic models on the level of energy flow rates and an hourly time scale for answering real-world questions concerning the optimal operation of energy supply systems. Moreover, we transfer the steady-state assumptions commonly used in the design optimization of energy supply systems to their operation, compare (*iii*), and investigate the impact of this simplification on the optimal solution.

As a compromise between computational costs and the quality of the solution of a parameter estimation problem, local optimization algorithms or stochastic global optimization algorithms and metaheuristics [e.g., 80, 153] are typically used for real-world applications [1, 19, 29, 59, 160, 163]. However, both local optimizer and metaheuristics can be trapped in suboptimal local solutions of multi-modal nonconvex models, compare (*ii*). Only deterministic global optimization methods are guaranteed to find the minimal possible model-data mismatch and, thus, allow to conclude from a large model-data mismatch that the model is unsuitable to predict the given dataset [140, 187]. Thus, we enhance a common deterministic global optimization approach by exploiting the general structure of parameter estimation problems for the solution of real-world estimation problems. Furthermore, we identified methods from statistics and machine learning aiming at the improvement of model predictions based on a reduced dataset. On the one hand, resampling is commonly used for reducing the bias and variance of estimators, see [58, 159] and [93, Chapter 5]. On the other hand, out-of-sample estimation is used

to obtain a better generalization of the optimized model to unseen data by fitting data points not used within training, the so-called validation set. Both of these approaches allow to enhance the global deterministic optimization of parameter estimation problems when transferred to the setup of this thesis, compare (*iii*).

This thesis is structured into two main parts, namely Chapter 2 about the efficient optimization of energy supply systems and Chapter 3 about a novel approach for global parameter estimation. For both of these parts, we provide a dedicated introduction and closing section as both chapters may be read independently. Finally, we stress the connection between the two parts once again in Chapter 4 which contains overall conclusions and future work.

2. Efficient optimization of dynamic energy systems

2.1. Introduction

Energy supply systems which are optimal even in presence of uncertainties can only be obtained when optimizing both design and operation simultaneously [e.g., 127, 134]. However, design optimization commonly applies quasi-steady models for the underlying operation to make the optimization problem computationally tractable for large energy systems [48, 70, 81, 120, 219], although a more reliable optimization of design and operation may require the use of dynamic models for the operation [66, 92, 130, 201].

While in principle deterministic global methods for the solution of nonconvex dynamic optimization problems exist for two decades [187], these are far from being applicable to large or even moderately sized energy systems coupling various energy conversion components, types of energy demands, and energy resources. An alternative are heuristic local methods such as multistart [e.g., 77]. However, a good start solution is typically required for convergence of the optimizer. In this chapter, we find the start solution based on ad-hoc adaptations of intermediate optimization results.

The optimization of design and operation of large-scale energy systems becomes tractable when formulating the optimization problem as a mixed-integer nonlinear program (MINLP) by using steady-state assumptions or even a mixed-integer linear program (MILP) by subsequent linearization, compare [128]. Indeed, quasi-steady models may be sufficient for modeling energy systems due to the fast response of the system components relative to the discretization of the demand time series [e.g., 11]. Apart from this, the presence of energy or material storages may be a sufficient justification for quasi-steady operation [50, 177]. However, considering the system-inherent dynamics may still be necessary, e.g., to obtain correct peak loads and, thus, the optimal dimensioning of the system components [45, 155, 208, 217, 220], although quasi-steady models may give first results [42].

The remainder of this chapter is structured as follows. In Section 2.3, we present standard dynamic models for boiler and combined heat and power units, see Subsection 2.3.1, and a thermal energy storage, see Subsection 2.3.2, at the level of heat transfer and power rates based on the quasi-steady models of Voll et al. (2013) [209]. Similar to [42], we reduce a complex dynamic model or, in reverse direction, extend a given system-level model to find a good compromise between accuracy and computational as well as modeling effort. Subsequently, we demonstrate the applicability of these models to answer relevant questions concerning the optimal operation of energy supply systems with a case study in Subsection 2.3.3. In Section 2.4, we investigate the characteristic properties of dynamic models to determine whether steady-state assumptions give sufficiently accurate optimization results. As a first theoretical step, we identify in Subsection 2.4.1 the system-inherent dynamics as

first-order dynamics depending on characteristic time constants. Consequently, we study the impact of different time constants on the operation of a boiler in both theory and practice. In Section 2.4.2, we complement our findings with a numerical case study on the optimal operation of a multi-component energy supply system for fulfilling a given heating and electricity demand, before we conclude in Section 2.5. Note that similar dynamic models with first-order dynamics have been exclusively used for boiler systems [186, 218] and a building model [46], while we apply first-order dynamics to all components of the energy system. As a consequence, the findings of this chapter can be transferred to similar systems with first-order dynamics stemming from other applications.

2.1.1. Data availability statement

For the dynamic optimization, we use our open-source solver DyOS available at https://git.rwth-aachen.de/avt-svt/public/dyos (Last accessed: Dec 1, 2024).
A ready-to-use implementation of energy supply system ES1 as well as the corresponding input time series are openly accessible via repository HECI, see https://git.es2050.org/heci/energy-benchmark (Last accessed: Dec 1, 2024).

2.2. Background to dynamic optimization

With *dynamic optimization* we refer generally to the solution of mathematical programs with differential equations embedded. This includes (*i*) optimizing finitely many unknown optimization variables and (*ii*) optimizing infinite dimensional functions, the so-called *controls*. The second class of optimization problems splits further into cases where (*ii.i*) the controls are independent of the process output and (*ii.ii*) a feedback of the current system output is used for determining the controls. Case (*ii.i*) is known by *optimal control*, dynamic offline optimization, or open-loop optimal/model-predictive control, while case (*ii.ii*) is often called model-predictive control, dynamic online optimization, real-time optimization, or closed-loop optimal/model-predictive control. In this thesis, we account for (*i*) finite-dimensional dynamic optimization, namely for the parameter estimation for dynamic systems, cf. Section 3.2.1, and for (*ii.i*) optimal control, namely for the dynamic optimization of the operation of energy supply systems.
In this chapter, we solve optimal control problems of the form

$$\min_{\boldsymbol{u}(\cdot),\,\boldsymbol{p}} \int_0^{t_f} L(\boldsymbol{x}(t),\boldsymbol{z}(t),\boldsymbol{u}(t),\boldsymbol{p})\,\mathrm{d}t \tag{OCP}$$

$$\begin{aligned}
\text{s.t. } \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}(t) &= f(\boldsymbol{x}(t),\boldsymbol{z}(t),\boldsymbol{u}(t),\boldsymbol{p}) && \forall t\in[0,t_f]\\
\boldsymbol{x}(0) &= \boldsymbol{x}_0 &&\\
0 &= \boldsymbol{h}^{\mathrm{eq}}(\boldsymbol{x}(t),\boldsymbol{z}(t),\boldsymbol{u}(t),\boldsymbol{p}) && \forall t\in[0,t_f]\\
0 &\geq \boldsymbol{h}^{\mathrm{ineq}}(\boldsymbol{x}(t),\boldsymbol{z}(t),\boldsymbol{u}(t),\boldsymbol{p}) && \forall t\in[0,t_f]\\
\boldsymbol{x}(t) &\in [\boldsymbol{x}^{\min},\boldsymbol{x}^{\max}] && \forall t\in[0,t_f]\\
\boldsymbol{u}(t) &\in [\boldsymbol{u}^{\min},\boldsymbol{u}^{\max}] && \forall t\in[0,t_f]
\end{aligned}$$

with n_u control functions $\boldsymbol{u} : [0,t_f] \to \mathbb{R}^{n_u}$ with $\boldsymbol{u}^{\min},\boldsymbol{u}^{\max} \in \mathbb{R}^{n_u}$ and n_p unknown parameters $\boldsymbol{p} \in \mathbb{R}^{n_p}$ as well as n_x differential states $\boldsymbol{x} : [0,t_f] \to \mathbb{R}^{n_x}$ with $\boldsymbol{x}^{\min},\boldsymbol{x}^{\max} \in \mathbb{R}^{n_x}$

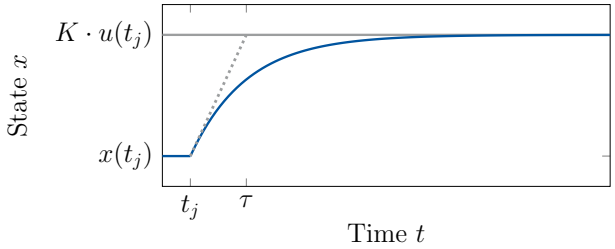


Figure 2.1.: Trajectory of state x (–) responding with first-order dynamics to a step function of height $u(t_j)$ triggered at time point t_j as well as the characteristic slope (· ·) at the initialization of the step and the characteristic asymptote (–)

and n_z algebraic variables $\boldsymbol{z} : [0, t_f] \to \mathbb{R}^{n_z}$ over time horizon $[0, t_f]$. The residuals of the equality and inequality constraints $\boldsymbol{h}^{\text{eq}}$ and $\boldsymbol{h}^{\text{ineq}}$, respectively, are continuous real-valued functions which are evaluated row-by-row. Note that the algebraic variables may be intermediate variables determined by the controls and differential states or a-priori given time series.

In particular, we focus on differential states with first-order dynamics [e.g., 149]

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = \frac{1}{\tau}(K \cdot u(t) - x(t)) \ ,$$

where the speed of the step response is characterized by time constant τ and the asymptote of the step response by steady-state gain K and input signal u, see Figure 2.1.

We use our open-source solver DyOS [41] for solving (OCP) which implements direct shooting methods. In the case studies of the following sections, we use single shooting [36, 115, 169, 189]. In detail, we use a piecewise constant discretization to transform the control functions into finitely dimensional control vectors and solve the resulting dynamic optimization problem by iterating between an integrator and an optimizer. The integrator solves the dynamic system on time horizon $[0, t_f]$ based on the current values of the optimization variables, in particular, of the discretized control vector, while the optimizer updates the optimization variables based on the state and sensitivity values provided by the integrator.

Another shooting method is multiple shooting [27], where the time horizon $[0, t_f]$ is split in shooting intervals $[t_i, t_{i+1}]$, $i = 0, \ldots, n_s - 1$ with $0 = t_0 < t_1 < \ldots < t_{n_s} = t_f$. In each shooting interval, the integration is initialized from an auxiliary optimization variables $\boldsymbol{s}_i$, $i = 0, \ldots, n_s - 1$. To obtain nevertheless a continuous state trajectory, matching constraints $\boldsymbol{s}_i = \boldsymbol{x}(t_i)$, $i = 0, \ldots, n_s - 1$ are added to the optimization problem. As for single shooting, an integrator and an optimizer are called iteratively for calculating the optimal solution.

Note that we reformulate the Lagrange-type objective of (OCP) into the equivalent Mayer-type objective for the use of DyOS.

2.3. Dynamic models for energy systems based on heat transfer rates

2.3.1. Dynamic models of boiler and CHP units

We extend the quasi-steady models used by Voll et al. (2013) [209] to obtain concise dynamic models for a boiler, a combined heat and power (CHP) unit, and a thermal energy storage (TES). The dynamic models are based on the time-dependent energy balance

$$\frac{\mathrm{d}T}{\mathrm{d}t}(t) = \frac{1}{c_p m}\left(\dot{Q}^{\mathrm{in}}(t) - \dot{Q}^{\mathrm{out}}(t) - \dot{Q}^{\mathrm{loss}}(t)\right)$$

neglecting any spatial changes of temperature T within each component of the energy system, where c_p and m are the heat capacity and mass, respectively, of the working fluid and $\dot{Q}$ describes the interchanged heat transfer rates. We can formulate the component models in sole dependence of heat transfer rates when applying the following assumptions.

Assumption 1. We use lumped element models with a single average value for any state variable for each component of the energy system.

Assumption 2. Let the following hold for any boiler and CHP unit.

(i) All heat losses are proportional to the input heat transfer rate $\dot{Q}^{\mathrm{in}}$ via thermal efficiency η^{th}. For simplification and comparability to the quasi-steady models of [209], we apply steady-state efficiency $\eta^{\mathrm{th}}(t) := \dot{Q}^{\mathrm{out}}(t)/\dot{Q}^{\mathrm{in}}(t)$ for any considered time point t in the dynamic models.

(ii) The output heat transfer rate $\dot{Q}^{\mathrm{out}}$ is proportional to thermal energy E via time constant τ

$$\dot{Q}^{\mathrm{out}}(t) = \frac{E(t)}{\tau}$$

for any considered time point t, where thermal energy E is measured with respect to the minimal temperature in the energy system as a reference.

With Assumptions 1 and 2 the energy balance for boiler and CHP units yields

$$\frac{\mathrm{d}\dot{Q}^{\mathrm{out}}}{\mathrm{d}t}(t) = \frac{1}{\tau}\left(\eta^{\mathrm{th}}\left(l(t)\right)\cdot\dot{Q}^{\mathrm{in}}(t) - \dot{Q}^{\mathrm{out}}(t)\right) \qquad \forall t \tag{2.1}$$

with time constant τ, part-load fraction

$$l(t) = \frac{\dot{Q}^{\mathrm{out}}(t)}{\dot{Q}^{\mathrm{nom}}} \qquad \forall t\ ,$$

and nominal thermal capacity $\dot{Q}^{\mathrm{nom}}$. Note that we exclude flow reversal via constraints $\dot{Q}^{\mathrm{in}}(t) \geq 0$ and $\dot{Q}_i^{\mathrm{out}}(t) \geq 0$.
As in [209], we assume that the efficiency η^{th} is provided for any boiler and CHP unit by the manufacturer or based on measurement data as a function of part-load fraction l, compare Assumption 2(i). In particular, we do not consider that the surface to volume ratio decreases with increasing size. Apart from this, we assume independency from

temperature for the efficiency as was claimed for conventional boilers in Augenstein et al. (2005) [12]. These assumptions about the efficiency pose major differences to the models used for simulating the dynamic operations of (components of) energy systems, e.g., in Modelica libraries AixLib [132], HKsim [217], TIL [162], and ThermoPower [40]. Therein, the efficiency is implicitly given via thermodynamic equations of heat and/or pressure losses.
Time constant τ can be interpreted as residence time of the working fluid within the respective system component and is therefore a measure for the "velocity" of the thermal inertia of the component. When further identifying efficiency η^{th} as gain in terms of control theory, (2.1) can be identified as first-order dynamics in the time domain. A similar approach using first-order dynamics in the frequency domain has been validated on the basis of experimental data for a boiler system [218].
Neither additional equations nor variables are necessary in dynamic model (2.1) compared to the quasi-steady model of [209]. Only a constant parameter is introduced, namely time constant τ, and algebraic variable $\dot{Q}^{\text{out}}$ is transformed into a differential state. In particular, the sparsity of the system matrices is only changed by the coupling of time steps which is necessary for introducing dynamics.

2.3.2. Dynamic model of a thermal energy storage

For the TES model, we adapt Assumption 2 as follows.

Assumption 3. Let the following hold for any TES unit.

(i) All heat losses are proportional to the stored thermal energy E.

(ii) Heat transfer rate $\dot{Q}_{\text{TES}}$ exchanged between TES and the remaining energy system is only bounded by the capacity of the storage and time constants.

For the heat losses, cf. Assumption 3(i), we consider (i) self-discharge caused by insufficient insulation and (ii) losses when discharging and charging the storage. While we model (i) in dependence of stored thermal energy E and a time constant τ^{loss} characterizing the rate of the energy loss, we use for (ii) the same constant efficiency $\eta = 0.95$ for both charging and discharging as in [18]. Thus, we obtain energy balance

$$\frac{\mathrm{d}E}{\mathrm{d}t}(t) = \eta \cdot \dot{Q}_{\text{TES}}(t) - \frac{1}{\tau^{\text{loss}}} \cdot E(t) \qquad \forall t$$

with heat transfer rate

$$\dot{Q}_{\text{TES}}(t) := \dot{Q}^{\text{in}}(t) - \dot{Q}^{\text{out}}(t) \qquad \forall t \ . \tag{2.2}$$

Note that only redundant solutions are eliminated by (2.2) as long as the input and output heat transfer rates are weighted with the respective efficiencies when accounting for different losses for charging and discharging. If the charging and discharging efficiency differ, a nonsmoothness would be introduced at the point of flow reversal. This is a general known issue for batteries, which is commonly tackled by the introduction of mixed-integer or MPEC's (Mathematical Programs with Equilibrium Constraint) [e.g., 145].

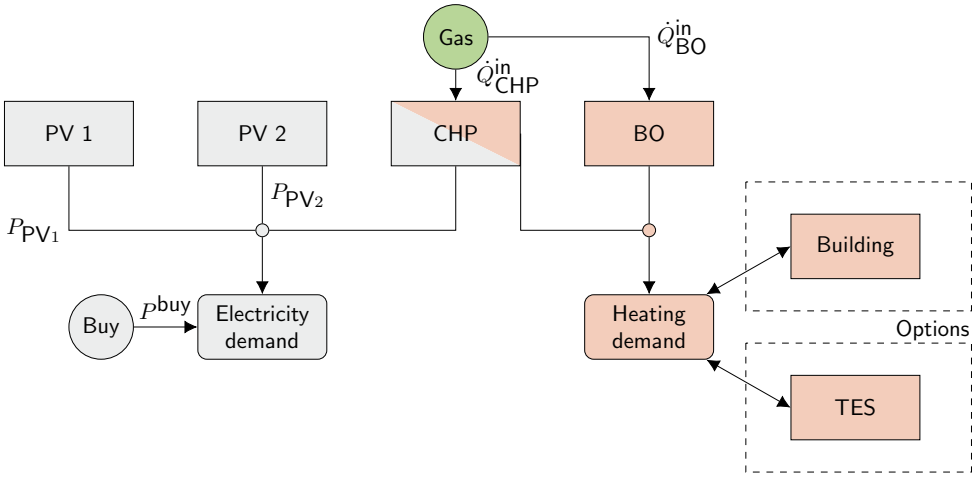


Figure 2.2.: Components and control variables of energy system ES1. In the simulation, the thermal energy storage TES is replaced by the implicit storage capabilities of building OB 1.

Table 2.1.: Characteristic parameters of the components of energy system ES1

Parameter			CHP	BO	TES
Nominal capacity	$\dot{Q}^{\text{nom}}$	[kW]	470	530	100
Minimum part-load fraction	l^{min}	[–]	0.5	0.2	–
Time constant	τ	[h]	0.1	0.1	1
Time constant for self-discharge	τ^{loss}	[h]	–	–	200

While input and output heat transfer rates of the boiler and CHP units have an explicit functional dependency, see Assumption 2(i), the heat transfer rates exchanged with the TES can take any value within the storage capabilities

$$-\frac{E(t)}{\tau} \leq \dot{Q}_{\text{TES}}(t) \leq \frac{E^{\text{nom}} - E(t)}{\tau} \qquad \forall t$$

with nominal capacity E^{nom}

$$0 \leq E(t) \leq E^{\text{nom}} =: \tau \cdot \dot{Q}^{\text{nom}} \qquad \forall t$$

and rate limit τ characterizing the maximum velocity of the working fluid due to tube dimensions and pressure restrictions, see also [18].
Note that, if applicable, the TES model may be extended, e.g., by considering further losses or different temperature layers [18, 179, 192].

2.3.3. Case study on the optimal operation of dynamic energy systems

We illustrate the applicability of the presented dynamic models for gaining insights into the optimal operation of energy supply systems with a case study. In fact, we compare the explicit storage capabilities of a thermal energy storage with the implicit storage capabilities of a building. We optimize the operation of the energy supply system *ES1* comprising

photovoltaic (PV) units, a CHP unit, and a boiler, see Figure 2.2, for fulfilling the heating and electricity demand of a research campus comprising eight office buildings and two experimental facilities, cf. Tables B.2 and B.3 in [170]. For the dynamic optimization, energy system ES1 includes a TES to allow for a more flexible operation of the boiler and CHP unit while still meeting the heating demand. In a subsequent simulation, we replace the TES with the thermal inertia of an office building. Thus, the size of the TES unit is adapted to the heat transfer capacity of this building, compare Table 2.1 and Appendix A.1.2. We use actual weather data, namely from Stuttgart in the winter week November 26, 2018 to December 02, 2018 [55], and the energy costs of a similar week during the year, namely time points 7 896 h - 8 063 h taken from [18], see Figure 2.3.

For the TES, boiler, and CHP unit we apply the models presented in Section 2.3, while each of the PV units is given by a time series of the solar irradiation utilizable by the respective buildings. The control variables are the heat transfer input of the boiler and the CHP unit, the power provided by the PV components, as well as the purchased electricity at any time point, namely hourly values for the considered week. The initial values of the part-load fractions of boiler and CHP are optimized. The TES is set to be initially half-full to allow for both charging and discharging. Rather than directly optimizing the heat transfer rate exchanged with the TES, we fix it to the difference between heating demand and the heat transfer rate provided by both the boiler and the CHP unit. We minimize the total costs subject to constraints on the utilizable solar irradiation as well as on the capacities of each component, cf. Table 2.1. Note that violations of path constraints may occur between discretization points, compare [72]. Besides, we do not impose periodic boundary conditions to not further restrict the optimizer. Since weather is not week-periodic, we cannot expect periodic operation to be optimal. Other researchers, e.g., [77] have imposed periodic boundary conditions to avoid discharging the storage, but periodic boundary conditions and oscillating systems bring substantial challenges [215]. The complete model formulation is provided in Appendix A.1.1.

For the dynamic optimization of the operation, we employ our in-house open-source dynamic optimizer DyOS invoking local nonlinear optimizer SNOPT [78] and integrator IDAS from the SUNDIALS suite [74, 94]. We use a feasibility tolerance of 0.01 and an optimality tolerance of 0.001.

As shown by Figure 2.4a, the heating demand is mainly fulfilled by the CHP unit. Only when the demand exceeds the capacity of the CHP unit during the high demand periods of the workdays, i.e., from 0 h to approximately 120 h, the boiler is used, compare Figure 2.4c. In particular, the boiler is run with a part-load fraction of about 30 % to 40 % which already enables a highly efficient operation close to the nominal efficiency $\eta_{\mathrm{BO}}^{\mathrm{nom,th}} = 0.8$, cf. Figure A.1. The TES allows to exploit the bi-generation capabilities of the CHP unit. In fact, only the volatility of the sale costs of electricity can be exploited, since the gas costs are constant and purchase of electricity is not required. At times with higher sale costs for electricity, the CHP unit over-fulfills the heating demand to generate more electricity, compare, e.g., the peaks of the energy costs in intervals $[18\,\mathrm{h}, 24\,\mathrm{h}]$, $[114\,\mathrm{h}, 120\,\mathrm{h}]$, and $[150\,\mathrm{h}, 168\,\mathrm{h}]$ with the heat transfer rate exchanged with the TES depicted in Figure 2.4a. Note that the PV units contribute only marginally to the electricity supply due to the low amount of utilizable solar irradiation in the considered winter week, see Figure 2.4a.

Subsequently, we transfer the capability of the TES as explicit storage to a building as implicit storage. In particular, the optimized heat transfer rate interchanged with the TES $\dot{Q}_i,\ i \in \mathrm{TES}$ is added to the heating demand $\dot{Q}^{\mathrm{th,dem}}$. The internal temperature within the

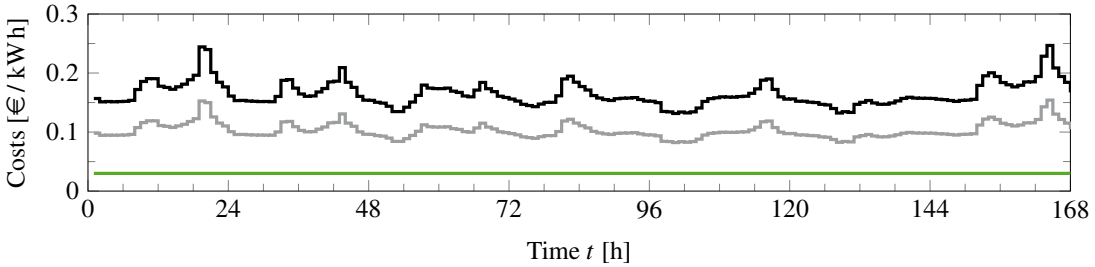
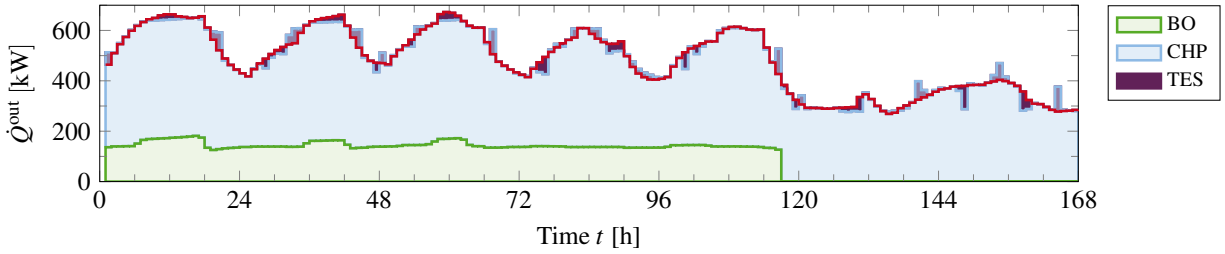
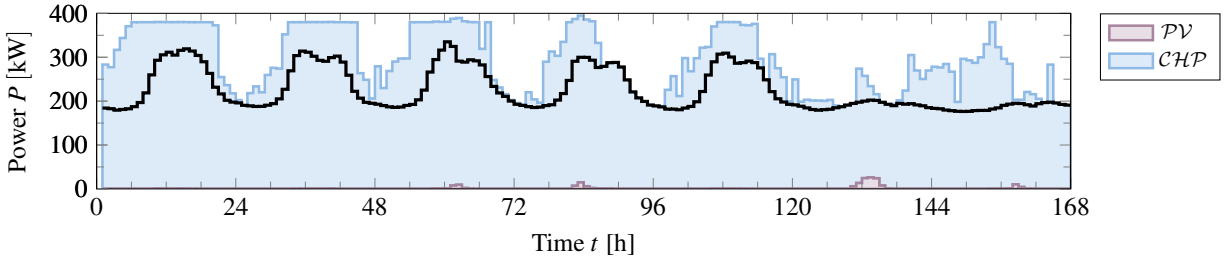


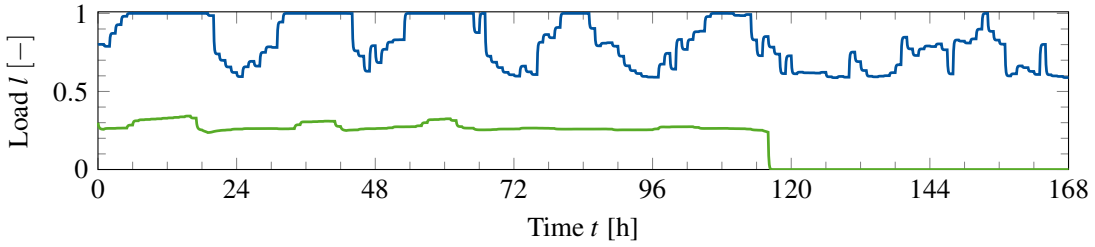
Figure 2.3.: Costs for buying (–) and selling (–) electricity as well as for buying gas (–) [18] underlying the operation of energy system ES1



(a) Satisfaction of heating demand (–) by CHP unit, boiler, and TES



(b) Satisfaction of electricity demand (–) by CHP and PV units



(c) Optimal part-load fraction of boiler (–) and CHP unit (–)

Figure 2.4.: Optimal operation of energy system ES1 for the heating and electricity supply of a research campus. Note that the shaded plots in 2.4a and 2.4b are additive.

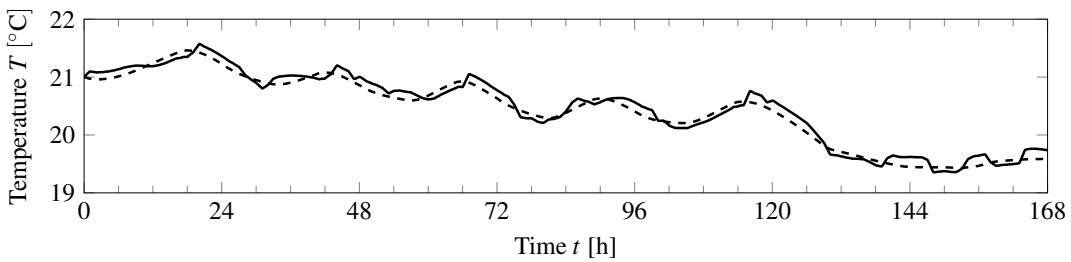
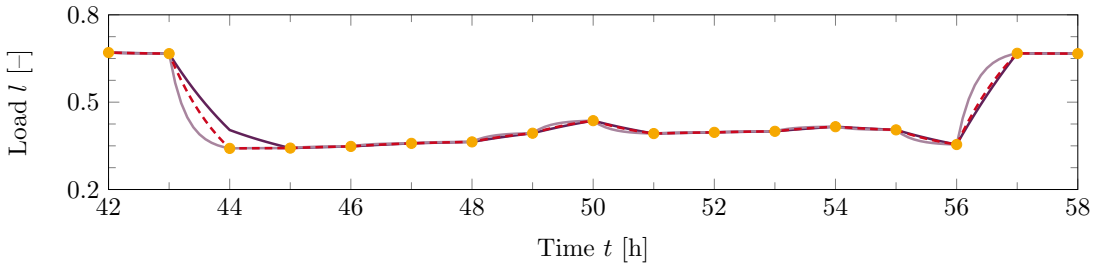
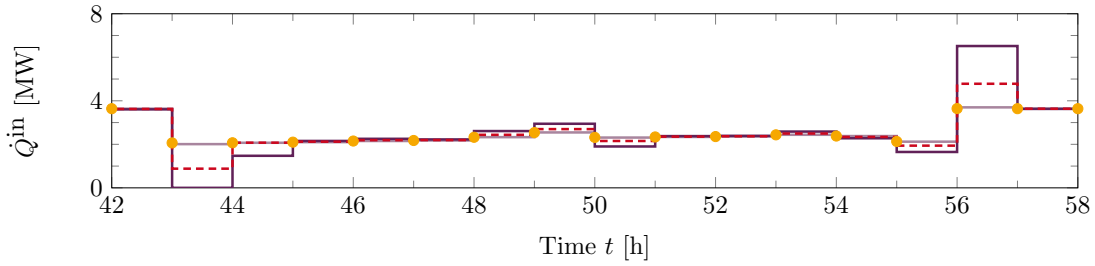


Figure 2.5.: Internal temperature in the building when using it as implicit storage (–) and when using a TES instead (- -)



(a) Differential state: part-load fraction



(b) Control: input heat transfer rate

Figure 2.6.: Optimal operation of a boiler with different dynamics, namely the dynamic model with $\tau = 2\,\text{h}$ (–), $\tau = 1\,\text{h}$ (- -), and $\tau = 0.25\,\text{h}$ (–) as well as with the quasi-steady model (●)

building is simulated based on heat demand $\dot{Q}^{\text{th,dem}}$, i.e., for energy system ES1 with an explicit storage, and the combined demand $\dot{Q}^{\text{th,dem}} + \dot{Q}_i,\ i \in \text{TES}$, i.e., for an analogous energy system with an implicit storage replacing the TES. Figure 2.5 shows that shifting the heat transfer rate $\dot{Q}_i,\ i \in \text{TES}$ from the TES to the building leads to visible but small oscillations in the range of up to 0.2 K.
The illustrative case study presented in this section shows that dynamic optimization helps to gain insights into the optimal operation of energy systems without the need to implement costly components just for testing purposes in reality. However, the optimization of energy systems considering many components is challenging for existing dynamic optimization software. Commonly, the dynamic models are therefore simplified with the help of steady-state assumptions. In the following section, we investigate to what extent such steady-state assumptions affect the optimal operation.

2.4. Quasi-steady versus dynamic models for energy systems

2.4.1. Characteristic properties of dynamic models

The standard models for the dynamic optimization of energy supply systems from Section 2.3 provide a well-tailored basis for investigating the impact of system-inherent inertia on the operation of the energy system: the models of each system component explicitly depend on a time constant τ which characterizes its delayed behavior. For first insights on the effect of different time constants on the optimal operation, we focus on the operation of one component, namely the boiler.
The optimization of the operation of a boiler unit reduces to a feasibility problem where

the output heat transfer rate is enforced to satisfy the heating demand

$$\dot{Q}^{\mathrm{th,dem}}(t) = \dot{Q}^{\mathrm{nom}} \cdot l(t) = \dot{Q}^{\mathrm{nom}} \cdot \left(l(0) + \int_0^t \frac{1}{\tau} \left(\eta^{\mathrm{th}}(l(\bar{t})) \cdot \frac{\dot{Q}^{\mathrm{in}}(\bar{t})}{\dot{Q}^{\mathrm{nom}}} - l(\bar{t}) \right) \mathrm{d}\bar{t} \right) . \tag{2.3}$$

Integration by substitution transforms (2.3) into

$$\dot{Q}^{\mathrm{th,dem}}(t) = \dot{Q}^{\mathrm{nom}} \cdot \left(l(0) + \int_0^{\bar{\tau}} \eta^{\mathrm{th}}(l(\bar{t})) \cdot \frac{\dot{Q}^{\mathrm{in}}(\bar{t})}{\dot{Q}^{\mathrm{nom}}} - l(\bar{t}) \, \mathrm{d}\bar{t} \right) \tag{2.4}$$

with boiler-specific time scale $\bar{\tau} := t/\tau$ characterized by "velocity" τ^{-1} [1/s]. The feasibility of path constraint (2.4) depends on the interrelation between the time scale of demand $\dot{Q}^{\mathrm{th,dem}}$ and the time scale of differential state l. The former time scale is characterized by step width $\Delta t > 0$ of the equidistantly chosen input discretization, the latter by time constant $\tau > 0$.

Figure 2.6 shows the control of the boiler for different interrelations of the time scales by varying time constant τ and simultaneously fixing step width Δt to 1 h. The underlying demand data are taken from [17].

The path constraint on the demand forces all state trajectories to the same values. Thus, the state trajectories can only deviate in the subintervals between the time points $\{\Delta t, 2 \cdot \Delta t, \ldots\} = \{1\,\mathrm{h}, 2\,\mathrm{h}, \ldots\}$ determined by the discretization of the path constraint, cf. Figure 2.6a. The faster the dynamics, i.e., the smaller time constant τ, the larger the possible deviations to the steady-state trajectory. While the characteristic convergence behavior of the first-order dynamics is clearly visible for the fastest dynamics (–), slower dynamics (–,- -) can hardly follow the necessary path constraints and, therefore, results in apparently piecewise linear trajectories.

An over- or undershooting of the control(s) may be necessary to obtain a feasible solution if large oscillations of state values are required within short time periods, cf. time intervals $[43\,\mathrm{h}, 45\,\mathrm{h}]$ and $[56\,\mathrm{h}, 57\,\mathrm{h}]$ in Figure 2.6b. The larger the required jump and the slower the underlying dynamics, the larger the deviation between subsequent control values. Conversely, the optimal control trajectory based on time constants significantly smaller than the discretization of the path constraint (–) coincides with the optimal quasi-steady control trajectory (•). Hence, explicit ramping constraints are necessary for models with relatively slow dynamics compared to the time discretization of the demand data when using quasi-steady state approximations to avoid unrealistic smooth control trajectories. In contrast to that, dynamic models intrinsically contain a ramping constraint via control bounds and inherent time constants. For example, the demand value at time point 44 h cannot be fulfilled by the slowest boiler unit (–) since the control value is bounded below to 0.

The suitable values for ramping constraints depend highly on the properties of the dynamic system. In particular, the change of the output of a system with first-order dynamics is limited by the height of the asymptote given by the gain as well as the speed of the convergence to this asymptote given by the time constant. In case of the considered boiler unit, this means that the change of the part-load fraction within an interval of width Δt is limited by the desired part-load fraction $\eta^{\mathrm{th}}(l(t)) \cdot \frac{\dot{Q}^{\mathrm{in}}(t)}{\dot{Q}^{\mathrm{nom}}}$ and time constant τ.

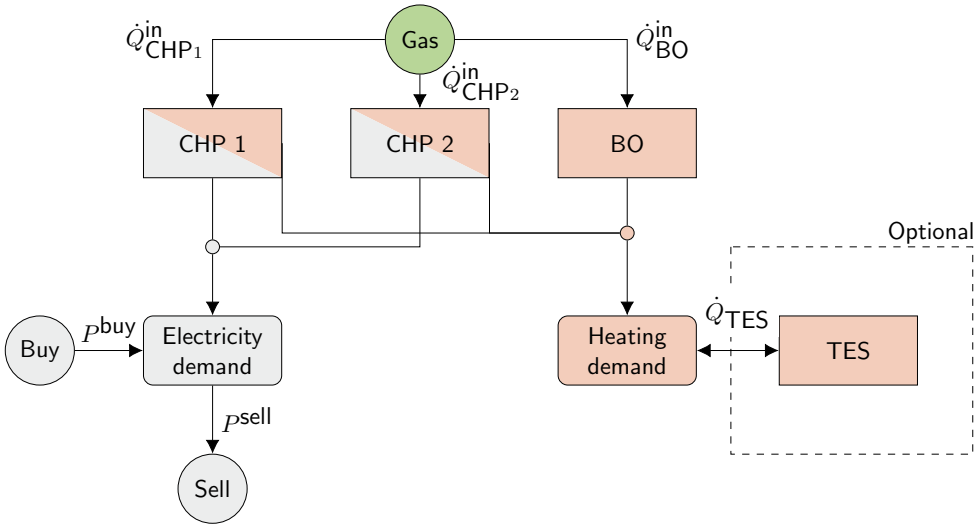


Figure 2.7.: Components and control variables of energy system ES2

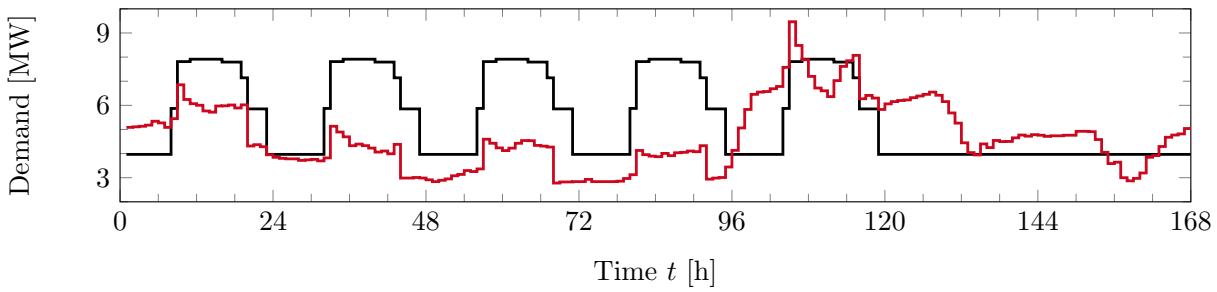


Figure 2.8.: Heating (–) and electricity (–) demand [17] underlying the operation of energy system ES2

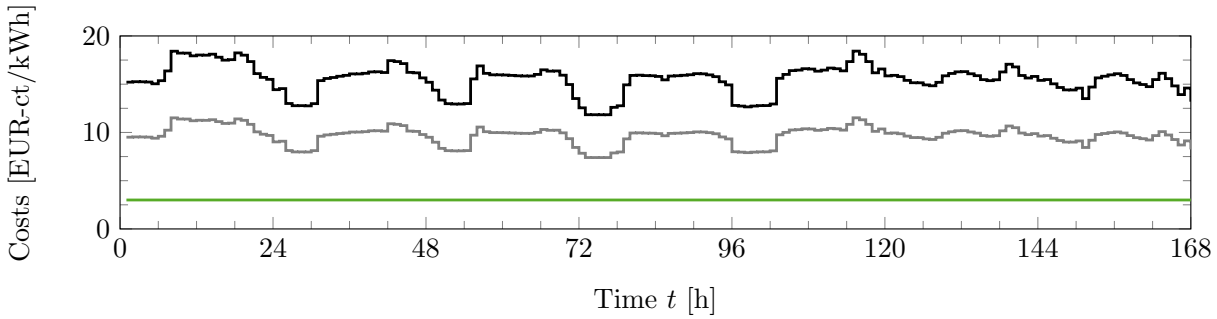


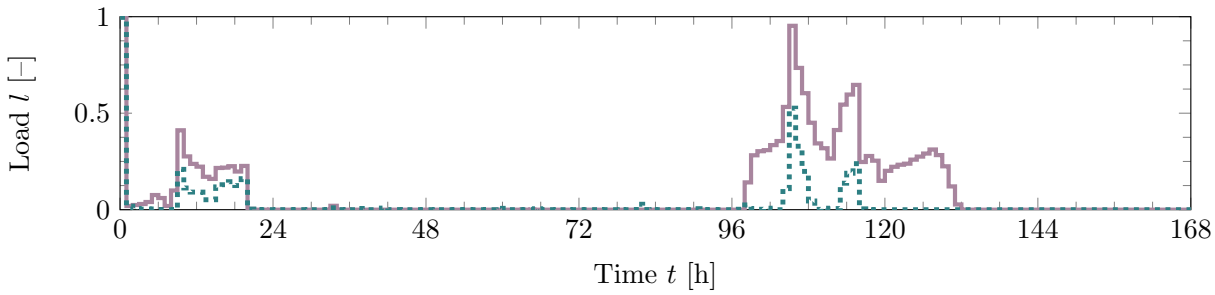
Figure 2.9.: Costs for buying (–) and selling (–) electricity as well as buying gas (–) [18] underlying the operation of energy system ES2

2.4.2. Case study on the operation based on different time delays

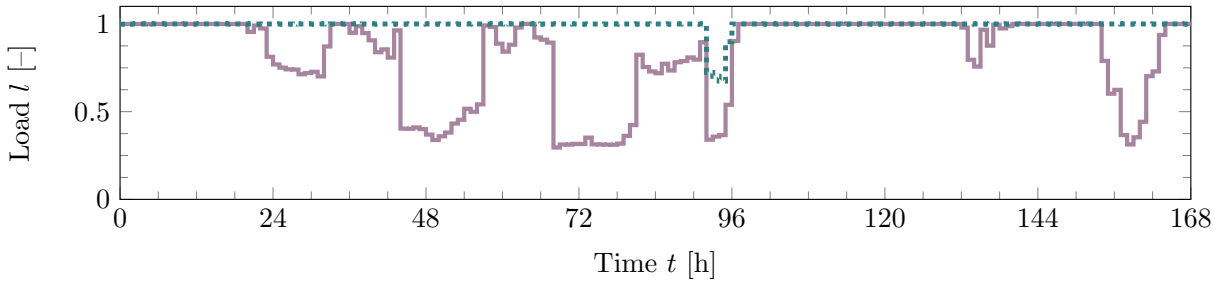
For studying the effect of different time constants on the operation of a multi-component energy supply system, we perform a case study with the energy supply system ES2 depicted in Figure 2.7, see also Appendix A.2. The operation of the energy system is optimized for one week, i.e., $t_f = 168\,\mathrm{h}$, on the basis of realistic demand data [17] as well as realistic electricity and gas costs [18], cf. Figure 2.8 and 2.9. Note that buying electricity just for selling it is uneconomical due to the substantial difference between purchase and sale costs. As was expected, our (local) optimal results indicate that electricity is only purchased if the demand cannot be fulfilled otherwise. The controls $\dot{Q}_i^{\mathrm{in}}(t)$, $i \in \{\mathrm{BO}, \mathrm{CHP1}, \mathrm{CHP2}\}$, $\dot{Q}_{\mathrm{TES}}(t)$, $P^{\mathrm{buy}}(t)$, and $P^{\mathrm{sell}}(t)$ for any time point $t \in [0, t_f]$ are equidistantly discretized into piecewiese constant functions with the same time step as the given data for demands and costs, namely $\Delta t = 1\,\mathrm{h}$ and $n_f = 168$.
The nominal capacities $\dot{Q}^{\mathrm{nom}}$ are chosen correspondingly to the provided demand data [17], namely $\dot{Q}^{\mathrm{nom}}_{\mathrm{CHP1}} = \dot{Q}^{\mathrm{nom}}_{\mathrm{CHP2}} = 2.5\,\mathrm{MW}$, $\dot{Q}^{\mathrm{nom}}_{\mathrm{BO}} = 4.5\,\mathrm{MW}$, and $E^{\mathrm{max}} = 100\,\mathrm{MW\,h}$. Since both maximum storage capacity $E^{\mathrm{max}}_{\mathrm{TES}}$ and time constant $\tau^{\mathrm{loss}}_{\mathrm{TES}}$ depend on the temperature range within the TES and its geometry, the time constant is approximated by $\tau^{\mathrm{loss}}_{\mathrm{TES}} = 333.3\,\mathrm{h} \approx 14$ days according to the chosen maximum storage capacity.
The actual time constants of boiler and CHP units are assumed to be in the range of one minute, which is covered by scenarios *steady* and *fast*, cf. Table 2.2. For comparison with other process systems engineering (PSE) applications, e.g., an air separation unit, a time constant of a few hours [89] is chosen as upper bound. On the one hand, these scenarios are considered without (capable) TES, meaning almost no storage capabilities due to slow response $\tau_{\mathrm{TES}} = t_f = 168\,\mathrm{h}$ in the dynamic cases and no TES at all in Scenario *steady*. On the other hand, an optimally responding TES is considered in scenarios with addendum "*+TES*" by using time constant

$$\tau_{\mathrm{TES}} = \frac{E^{\mathrm{max}}_{\mathrm{TES}}}{\max_{t\in[0,168]} \dot{Q}^{\mathrm{th,dem}}(t)} = \frac{100\,\mathrm{MW\,h}}{10\,\mathrm{MW}} = 1\,\mathrm{h}\ ,$$

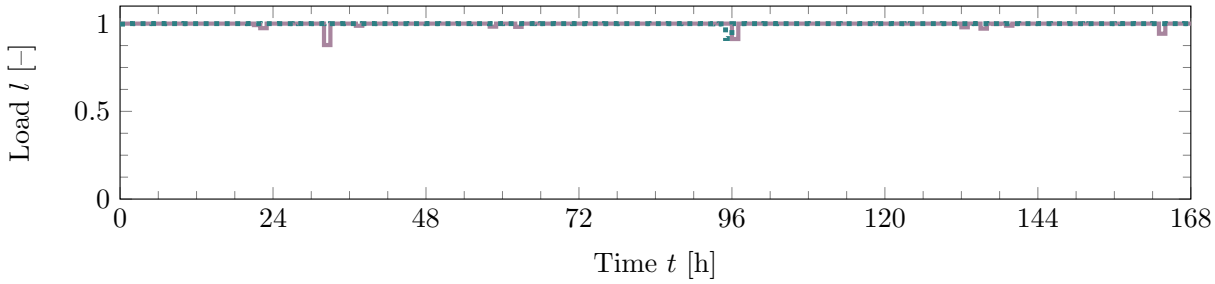
which means that the TES can fulfill the maximal hourly demand almost instantaneously. For the dynamic optimization, we use our open-source solver DyOS calling Ipopt [210] for solving the nonlinear optimization problem and NIXE [87] for the integration of the dynamic equation system. The quasi-steady solutions are calculated with Artelys Knitro 10.3.0 [38] invoked via GAMS 25.1.1 [73].
The optimal states are depicted in Figure 2.10 for scenarios *fast*(–) and *fast+TES* (··). In Scenario *fast*(–), the boiler and CHP units need to run such that they fulfill the heat demand almost instantaneously due to the slow response of the TES, whereby the operation of the CHP units is preferred to the operation of the boiler. This preference is fully exploited if a fast responding TES is present: the CHP units run most of the time on full load, while the load of the boiler unit is reduced significantly (··). Operating the CHP units at high load has two major advantages. On the one hand, the efficiency of the CHP units increases with increasing part-load fraction. On the other hand, the at-site electricity generation is cheaper than purchasing electricity, since the gas costs in year 2018 have been much lower than the electricity costs. Consequently, the storage is loaded to a small degree in both scenarios. Apart from this, no overproduction is favorable since electric power cannot be stored and selling over-produced electricity is not profitable enough.



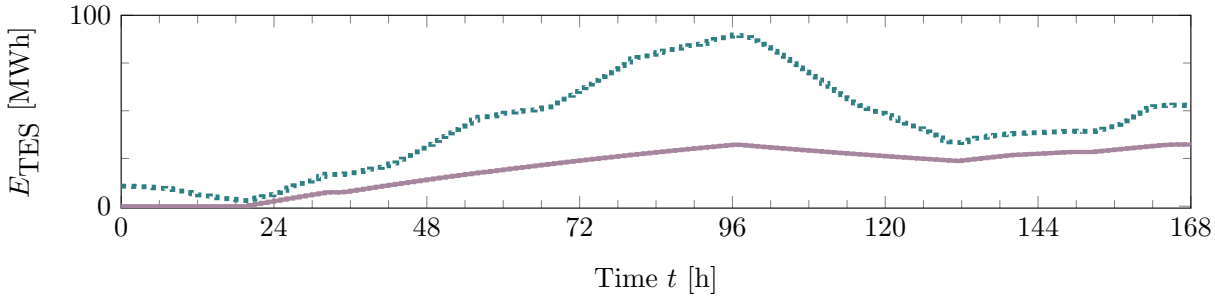
(a) Part-load fraction of boiler



(b) Part-load fraction of CHP 1



(c) Part-load fraction of CHP 2



(d) Storage level of TES

Figure 2.10.: Optimal state trajectories for scenarios *fast*(–) and *fast+TES* (··)

Scenario	$\tau_i,\ i \in \{\text{BO} \cup \text{CHP}\}$	Objective [EUR]	Rel. diff.[a] [–]
slow	2	–[b]	–
fast	0.25	97,526	0.00
steady	–	99,736	0.02
slow+TES	2	88,872[c]	0.04
fast+TES	0.25	88,792[c]	0.04
steady+TES	–	85,435	0.00

[a]Relative difference between the objectives with respect to the lowest objective found over all scenarios with/out a (capable) TES.

[b]Optimizer did not converge to feasible solution.

[c]Optimizer reached time limit. Feasible solution given.

Table 2.2.: Optimal objective values for the operation of energy system ES2 based on dynamic and quasi-steady optimization

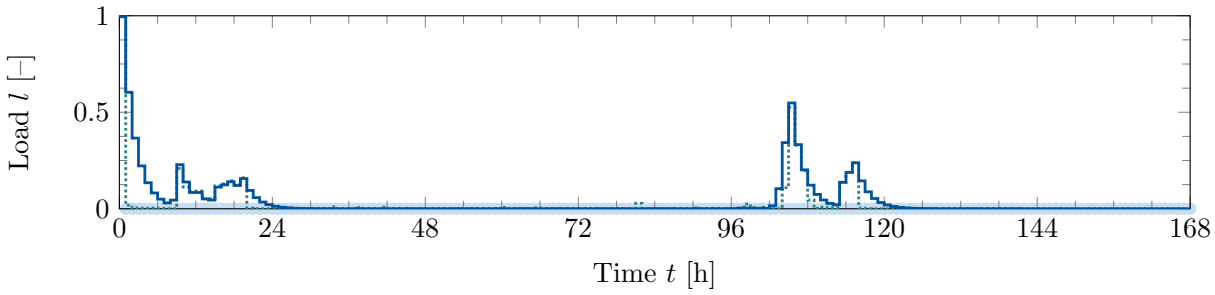


Figure 2.11.: Optimal part-load fraction of boiler for scenarios *slow+TES* (–), *fast+TES* (··), and *steady+TES* (•)

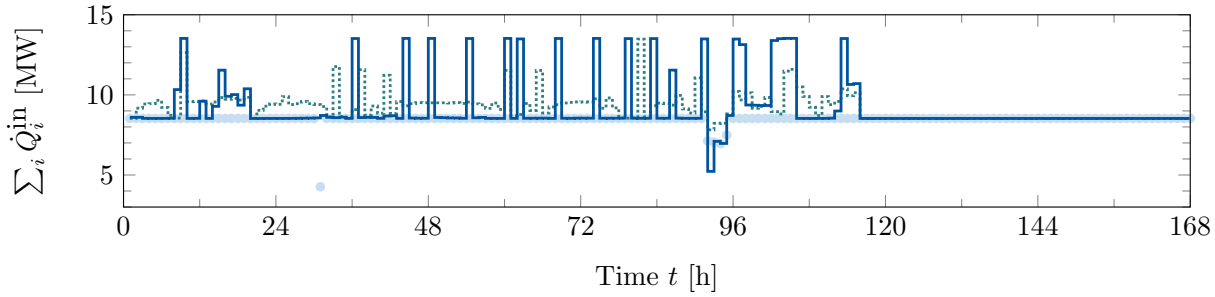


Figure 2.12.: Optimal total heating input, corresponding to the purchased gas, for scenarios *slow+TES* (–), *fast+TES* (··), and *steady+TES* (•)

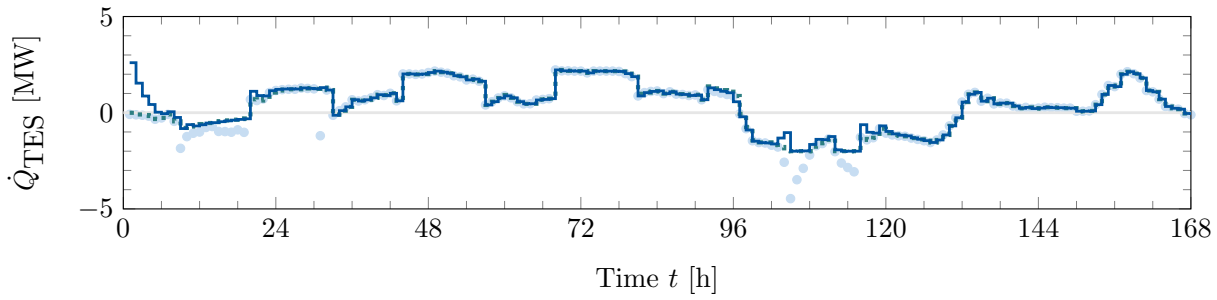


Figure 2.13.: Optimal trajectory of storage level for scenarios *slow+TES* (–), *fast+TES* (··), and *steady+TES* (•)

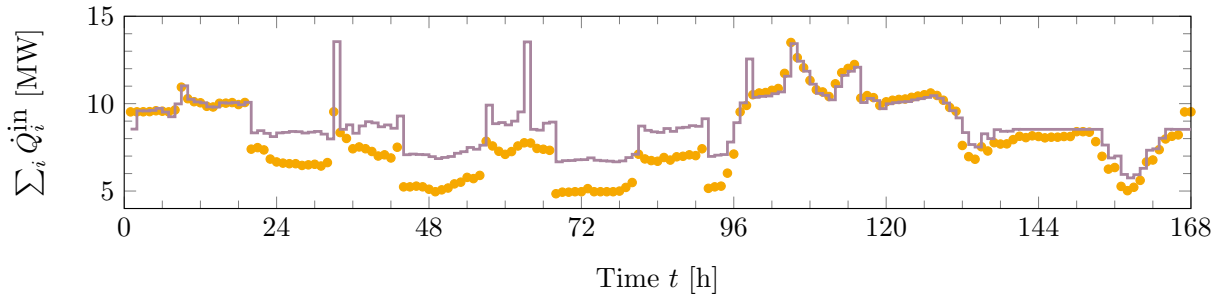


Figure 2.14.: Optimal total heating input, corresponding to the purchased gas, for scenarios *fast* (–) and *steady* (•)

The findings of Section 2.4.1 on the effect of different time constants on the optimal operation are confirmed by the case study. The dynamic state trajectories approach the quasi-steady solution with decreasing time constant, i.e., from Scenario *slow+TES* to Scenario *fast+TES*, cf. Figure 2.11. Furthermore, extreme control actions may be necessary if the components are characterized by slow dynamics, compare the oscillations of Scenario *slow+TES* (–) in Figure 2.12. Figure 2.12 as well as Figure 2.13 emphasize that dynamic and quasi-steady operation may vary significantly even if there is a storage unit. These differences could affect a design build on top of this operational results, since a boiler unit with lower capacity could be chosen in Scenario *fast+TES* compared to Scenario *slow+TES* and no boiler is necessary in Scenario *steady+TES*. However, the differences between the optimal trajectories hardly affect the weekly costs. The objective values of all scenarios with (capable) TES only vary within a range of 4%, see Table 2.2. Note that both the dynamic and the quasi-steady optimization problems contain several local optima that are close to each other. The local solution found as well as the corresponding computational runtime depends highly on the initial guess for the control values, whereby the quasi-steady optimization can be expected to be the fastest.

In the case without (capable) TES, the optimizer did not find a feasible solution for the dynamic optimization problem with large time constants, namely Scenario *slow*. Taking into account the difference between no and no capable TES, Scenarios *fast* and *steady* give similar results, cf. Table 2.2 and Figure 2.14. In particular, the dimensioning of the boiler and CHP units would not be affected by these differences if this optimization of the operation were the basis for a design optimization. Both Scenario *fast* and Scenario *steady* exploit the maximum capacity of the boiler and CHP units.

2.5. Conclusions and outlook

In this chapter, we derived concise dynamic models for different components of an energy supply system, namely a boiler, a CHP unit, and a TES, based on the quasi-steady models of Voll et al. (2013) [209]. We used these models in a brief case study on the optimal operation of an energy supply system comprising photovoltaic units, a CHP unit and a boiler for satisfying the energy demand of a generic research campus. Although we use a rather coarse scale for modeling, namely energy balances based on heat transfer rates and electric power, we obtain realistic results: The boiler is only used when the heating demand exceeds the capacity of the CHP unit, since the bi-generation capabilities of the CHP make its operation more economic. The TES unit allows for load shifting to exploit the volatile electricity prices. Finally, we compared the implicit storage capabilities of a building, due to its thermal inertia, with the explicit usage of a similarly sized TES unit resulting in only small temperature deviations.
Subsequently, we investigated whether quasi-steady state assumptions are valid for decreasing the computational complexity when optimizing the operation of energy supply systems. For this, we characterized the behavior of the presented component models as first-order dynamics. Consequently, we identified an explicitly given time constant as the characteristic property for the system-inherent dynamics. At first, we investigated the impact of dynamics on the operation of a single boiler. We showed that quasi-steady state approximations are adequate if the characteristic time constant is small compared to the discretization of the path constraint given by input data as long as sufficiently large control values are feasible in the dynamic case. These results apply analogously to other processes with first-order dynamics. Besides, the methodology of identifying characteristic time constants and varying them can be transferred to models with other types of dynamics to investigate whether quasi-steady models are adequate for optimizing their operations. Note that we have restricted ourselves to the impact of model properties on the optimal operation. Investigations on additional influence quantities given by the solution approach, e.g., the accuracy of the control grid when using control vector parametrization [115, 216], are left to future work.
We also studied the impact of varied dynamics on the optimal operation of a complex hybrid energy system taking into account time-varying input data for demands and prices. Our case study has confirmed the common finding from the literature that the extent of the dynamics mainly influences the peak values of the controls. Contrarily to common assumptions in the literature, our optimal dynamic and optimal quasi-steady operational strategies show significant differences even for fast dynamics, e.g., a time constant of 15 minutes compared to hourly demand and price deviations. As a consequence, quasi-steady state assumptions can be adequate for optimizing the operation of energy systems only if realistic ramping constraints are introduced. Moreover, our results demonstrate that quasi-steady state assumptions are not adequate just due to the presence of a storage unit. The findings for operational optimization with fixed design allow for some conjectures for iterative design optimization algorithms, where the operation of the currently active design is optimized in each iteration. In fact, two aspects of the optimal operational results mainly influence the evaluation of the currently active design: peak loads and operational costs. Since different peak loads may influence the choice of optimal nominal sizes, quasi-steady state assumptions may be adequate for a respective design optimization only if realistic ramping constraints are included. In contrast to that, the deviations of the operational

strategies in dependence of different time constants hardly affected the weekly operational costs of the energy system. In our case study, the costs were only increased by 4% when assuming overly pessimistic time constants of 2 hours compared to the minimum costs of the quasi-steady model in spite of the significant changes in the operational strategy.
Moreover, we may exploit methods and results from the integrated optimization of scheduling and operation of an energy supply system for the design optimization. For example, [158] investigate the effect on system-inherent inertia of energy supply systems on the optimal scheduling. In detail, they consider a delayed response of the heating output to setpoint changes as well as a static ramping constraint for the adjustment of the electricity output to the new setpoint for the integrated CHP unit. With this, the optimal solution for scheduling can exploit the thermal inertia as implicit storage capabilities and gives more realistic results regarding the actual electricity output of the CHP unit. However, first-order dynamics and static ramping constraints may not be sufficient for covering high-order dynamics [14], see [15, 53, 110] for more flexible ramping constraints.
A detailed case study for quantifying the effect of system-inherent dynamics on the design optimization of energy supply systems is left for future work. Such a study would lead to the solution of MINLPs or nonsmooth optimization problems when accounting for zero-operation of boiler and CHP units below a given minimum part-load fraction and different efficiencies for charging and discharging a TES. In particular, the necessity of global optimal solutions for both the design and the underlying operation may be studied, since the investigated nonlinear dynamic optimization problem is characterized by multiple local optimal solutions. A fair comparison of quasi-steady and dynamic models requires the choice of adequate values for the respective model parameters, e.g., determining the ramping constraints, which can be obtained by the optimal solution of corresponding parameter estimation problems. Choosing a suboptimal local solution for any of the mentioned optimization problems may distort the evaluation of the underlying control strategy as much as assuming unrealistic dynamics. However, the size of MINLPs and (mixed-integer) dynamic optimization problems which can be solved by deterministic optimization methods is still limited. In the following chapter, we therefore propose a novel approach for making the global optimization of general large-scale parameter estimation problems tractable, including the solution of MINLPs and dynamic optimization problems.

3. Global parameter estimation exploiting reduced datasets

3.1. Introduction

3.1.1. Related literature

Parameter estimation is typically required to obtain accurate models for the simulation and optimization of real-world processes from industry, the prediction of the performance of drugs and therapies in medicine or various other applications [e.g., 7, 65, 75]. Often, the resulting optimization problems are challenging for deterministic global optimization (DGO) methods [33, 67, 96] as they account for large datasets, nonconvexities, and, potentially, for discrete optimization variables [e.g., 16, 114, 117]. Thus, we propose to exploit the structure of parameter estimation problems to enable their solution with a common DGO method, namely the branch-and-bound (B&B) algorithm [97, 195]. In particular, we introduce the *B&B algorithm with growing datasets*: We use a reduced dataset for bounding and, potentially, pruning nodes to save computational effort when being far away from the global solution, i.e., at the beginning of the B&B algorithm. Subsequently, we let the dataset grow, meaning we augment data points, until we converge to the originally provided dataset.

It is well known that data reduction allows for more sophisticated fitting techniques, e.g., for using nonlinear models rather than linear regression [123]. More specifically, our approach has similarities to the widely-known usage of stochastic gradient descent algorithms or batch gradient descent algorithms to make the training of large-scale models in machine learning [e.g., 39, 144] or appointment scheduling [151] tractable. However, while their approaches are used to speed up local optimization procedures, we obtain a DGO method. As another difference, the (mini)batches or single data points are regularly resampled, typically without changing the number of data points. Similarly, Agosti et al. [6] iteratively update their reduced order model without changing its size based on information from the full order model to solve a complex parameter estimation problem from medicine. In contrast to that, we increase the size of the reduced dataset gradually as done for infinite sum problems [see, e.g., 24] or in stochastic optimization [126, 165, 214]. While the latter two approaches cannot know the true final dataset, our final dataset is given by the dataset originally provided with the model. Moreover, the branch-and-lift algorithm [98] iteratively increases the number of control variables within a spatial B&B algorithm to converge to the global optimum of optimal control problems. While we test the performance of the B&B algorithm also for parameter estimation problems with differential equations embedded, we do not solve optimal control problems as we optimize over finitely many unknown parameters, cf. Section 2.2. Finally, the sample-augment algorithm [86] for the solution of network design problems starts from a reduced subproblem and augments the

capacities of the edges until finding a solution of the original problem. While this is a heuristic algorithm which aims at finding a good solution with reasonable computational effort, we propose an algorithm which is guaranteed to converge to the global optimum.
The spatial B&B algorithm is proven to converge to the global optimum in finite time by various groups. We extend the proof of convergence given in Theorem 5.26 of [131]. The proof of Locatelli and Schoen [131] is less general than the standard proof of Horst and Tuy [97], but considers using an optimality tolerance for the successful termination of the B&B algorithm. Note that Kirst et al. [112] give a proof of convergence extended to the use of numerical feasibility tolerances under assumptions covering the setup of [97].

3.1.2. Structure of the chapter

The remainder of this chapter is structured as follows. In Section 3.2 we give an overview on the general model formulation and notation as well as the spatial B&B algorithm used in this chapter. Section 3.3 provides the foundation for the proposed algorithm: Based on a numerical study, we show that reducing the dataset has the potential to reduce the computational effort for solving the lower and upper bounding problems while preserving the order of magnitude of the lower and upper bounds. See Appendix B for additional information. Subsequently, we introduce the B&B algorithm with growing datasets in Section 3.4. In detail, we discuss in Subsection 3.4.1 how to process the B&B nodes based on reduced datasets and propose in Subsection 3.4.2 different augmentation rules to decide when to augment the dataset. This is followed by the proof of convergence of the B&B algorithm with growing datasets to the global optimum in Subsection 3.4.3. Appendix C repeats definitions used by [131] within their proof of convergence in the notation of this thesis. Since the performance of the deterministic approach presented in Section 3.4 may suffer from the lower bounds calculated based on the reduced datasets being too loose, we propose to use a tighter approximation of the lower bound based on the original dataset for pruning in Section 3.5. This yields the SSE and MSE heuristics for both of which we propose a post-processing update of the final lower bound to compensate for potential mistakes made with the heuristic pruning, cf. Subsections 3.5.1 and 3.5.2. In Subsection 3.5.3 we briefly discuss how the heuristic pruning affects bound tightening heuristics. The performance of the standard B&B algorithm and of the different approaches of the B&B algorithm with growing datasets is investigated in the benchmark study in Section 3.6 based on 13 real-world parameter estimation problems, see Appendix D. Finally, we provide conclusions and perspectives on future work in Section 3.7.

3.1.3. Data availability statement

The proposed B&B algorithm with growing datasets is implemented in our open-source DGO solver MAiNGO available at `https://git.rwth-aachen.de/avt-svt/public/maingo` (Last accessed: Dec 1, 2024). The deterministic approach was first available in version 0.6.0 and the heuristic approaches in version 0.8.0.
Ready-to-use implementations as well as the corresponding full datasets of all parameter estimation problems used for numerical studies in this chapter are openly accessible via repository GloPSE, see `https://git.rwth-aachen.de/avt-svt/public/glopse` (Last accessed: Dec 1, 2024).

3.2. Background to global parameter estimation

3.2.1. General parameter estimation problem

We focus on solving the general parameter estimation problem

$$\begin{aligned} \min_{\boldsymbol{p}\in\mathcal{P}} \quad & \sum_{(\boldsymbol{x}_d,y_d)\in\mathcal{D}} g(\boldsymbol{p};\boldsymbol{x}_d,y_d) \\ \text{s.t. } & h(\boldsymbol{x}_d,y_d;\boldsymbol{p}) \leq 0 \qquad \forall d=1,\dots,|\mathcal{D}| \\ & \widetilde{h}(\boldsymbol{p}) \leq 0 \ , \end{aligned} \tag{PE}$$

where $\boldsymbol{p} \in \mathcal{P}$ with a closed, bounded box $\mathcal{P} \subsetneq \mathbb{R}^n$ are the unknown parameters, $\mathcal{D} = \{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\dots\} \subsetneq \mathbb{R}^m \times \mathbb{R}$ denotes the full set of measurement data, $h(\boldsymbol{x}_d,y_d;\cdot) : \mathcal{P} \to \mathbb{R}$ is the residual of an inequality constraint in dependence of a single data point $(\boldsymbol{x}_d,y_d) \in \mathcal{D}$, and $\widetilde{h} : \mathcal{P} \to \mathbb{R}$ is the residual of a data-independent inequality constraint on the parameters. Note that we restrict ourselves to one constraint $h(\boldsymbol{x}_d,y_d;\cdot)$ and $\widetilde{h}$ only to keep the notation simple. All theoretical statements follow analogously for multiple constraints with the same general properties.
The objective function is composed of non-negative data-dependent functions $g(\cdot;\boldsymbol{x}_d,y_d) : \mathcal{P} \to \mathbb{R}_{\geq 0}$ for any data point $(\boldsymbol{x}_d,y_d) \in \mathcal{D}$. In detail, we minimize the summed squared prediction error

$$g_{\mathcal{D}}(\boldsymbol{p}) := \sum_{(\boldsymbol{x}_d,y_d)\in\mathcal{D}} g(\boldsymbol{p};\boldsymbol{x}_d,y_d) \tag{SSE}$$

of a general parameterized model function $f(\cdot;\boldsymbol{p}) : \mathbb{R}^m \to \mathbb{R}$ based on least-squares deviations

$$g(\boldsymbol{p};\boldsymbol{x}_d,y_d) := \left(f(\boldsymbol{x}_d;\boldsymbol{p}) - y_d\right)^2$$

for each data point $(\boldsymbol{x}_d,y_d) \in \mathcal{D}$ and parameters $\boldsymbol{p} \in \mathcal{P}$. We call (PE) the *original problem* and, when replacing the full dataset with a reduced dataset $\mathcal{D}_k \subseteq \mathcal{D}$, the *reduced problem*. Consequently, we define a parameter estimation problem via its model, i.e., mathematical formulation (PE), and its dataset.
For the theoretical considerations in Sections 3.4 and 3.5, we focus on continuous-valued optimization parameters $\boldsymbol{p} \in \mathbb{R}^n$ as done by [131]. In contrast to that, we extend (PE) to MINLPs in the numerical case study. The parameter domain $\mathcal{P}$ of unknown integer parameters $\boldsymbol{p} \in \mathbb{Z}^{n_d}$ is given by subsequent discrete values. Moreover, we can solve dynamic optimization problems (DOs) by applying discretize-then-relax methods [e.g., 60, 138, 140]. In particular, we transform any differential equation into an algebraic equation using the explicit Euler scheme. Besides, we can easily extend (PE) for model functions $f(\cdot;\boldsymbol{p}) : \mathbb{R}^m \to \mathbb{R}^{n_f}$ with higher dimensional output $n_f > 1$ by using the Euclidean norm to measure the prediction error

$$g(\boldsymbol{p};\boldsymbol{x}_d,y_d) = ||f(\boldsymbol{x}_d;\boldsymbol{p}) - y_d||^2 \ \text{ for } \boldsymbol{p} \in \mathcal{P},\ (\boldsymbol{x}_d,y_d) \in \mathcal{D} \ .$$

3.2.2. Spatial B&B algorithm

For solving (PE) to global optimality, we apply the spatial B&B algorithm [e.g., 96, 195], which we also call the *standard* B&B algorithm in the following. The B&B algorithm

Algorithm 1 Standard B&B algorithm

Input: Model formulation including domain of optimization variables $\mathcal{P}$, optimality tolerance ε

Output: Global optimal solution including bounds on optimal objective value

1: Initialize trivial bounds $u_0 := \infty$ and $l := -\infty$ as well as counter $k := 0$, domain of root node $\mathcal{P}_0 := \mathcal{P}$ and set of active nodes $\mathcal{N}_0 := \{N_0\}$;
2: Pre-processing of root node N_0 ;
3: **while** $u_k - l > \varepsilon$ **do**
4: Select next node $N_k \in \mathcal{N}_k$;
5: Solve lower bounding problem in N_k ;
6: Solve upper bounding problem in N_k ;
7: Update best upper bound found so far u_k ;
8: **if** N_k cannot be pruned **then**
9: Branch N_k by partitioning $\mathcal{P}_k = \mathcal{P}_{k,1} \cup \mathcal{P}_{k,2} \cup \dots$;
10: $\mathcal{N}_{k+1} := (\mathcal{N}_k \setminus N_k) \cup N^{\mathrm{k},1} \cup N^{\mathrm{k},2} \cup \dots$;
11: **else**
12: $\mathcal{N}_{k+1} := \mathcal{N}_k \setminus N_k$;
13: **end if**
14: $\mathcal{N}^{\mathrm{f}}_{k+1} := \mathcal{N}^{\mathrm{f}}_k \cup N_k$;
15: Update lower bound over all active nodes l ;
16: $k := k + 1$;
17: **end while**

iteratively builds a tree structure to explore the feasible domain of the optimization variables, namely parameter domain $\mathcal{P}$ in case of (PE), for finding the global solution, see Algorithm 1. In each B&B iteration k, a node N_k of the tree is fathomed: At first, lower and upper bounds on the global solution value within this node are calculated (*Bounding*). If the lower bound of the node l_k is larger than the best upper bound u_k found in nodes $N_{\tilde{k}}$, $\tilde{k} \leq k$, the node is pruned as it cannot contain the global solution. Otherwise, the parameter domain $\mathcal{P}_k$ of the node is partitioned into disjoint subsets (*Branching*). The B&B algorithm converges if the best lower and upper bounds valid for all nodes of the B&B tree are closer than an a-priori given optimality tolerance ε.

For the upper bounding, mere function evaluations or local optimizations can be used. The lower bounding typically uses convex relaxations of the model, e.g., by exploiting knowledge on the relaxations of intrinsic functions in combination with the auxiliary variable method (AVM) [167, 190, 191, 195, 196], by introducing quadratic terms according to the αBB method [4, 5, 9], or by McCormick relaxations [136, 146, 200]. In common DGO software like ANTIGONE [139], BARON [111, 196], MAiNGO [31], and SCIP [3, 203], the resulting convex lower bounding problem is further relaxed into a linear program. Also, most DGO software implements bound tightening techniques [e.g., 79], since the fathoming of B&B nodes becomes more efficient when obtaining tighter upper and lower bounds.

In this thesis, we use and extend our in-house open-source solver MAiNGO which implements a spatial B&B algorithm based on McCormick relaxations obtained by MC++ [43] for the DGO of factorable optimization problems. We implemented a very basic integer treatment to make MAiNGO capable of solving MINLP's. In detail, we (*i*) round any non-integer domain bounds when branching discrete variables and (*ii*) relax the integrality

constraint for solving the lower and upper bounding problem. To ensure a feasible upper bound with (*ii*), the determined solution values of all discrete optimization variables is rounded to the nearest integer. Subsequently, we check the feasibility of the resulting solution point for the original problem and deprecate the solution point in case of infeasibility. We are sure to obtain a feasible solution point for the upper bounding problem or to prove the infeasibility of the problem at latest when all discrete optimization variables are fixed to integer values via rounding the domain bounds (*i*). Note that binary variables are fixed after the first branching in their parameter domain.

3.3. Effect of data reduction on the global parameter estimation

As a first step, we study how the reduction of data points in full dataset $\mathcal{D}$ affects the performance of the spatial B&B algorithm for a (specific) parameter estimation problem. In detail, we investigate the following hypotheses:

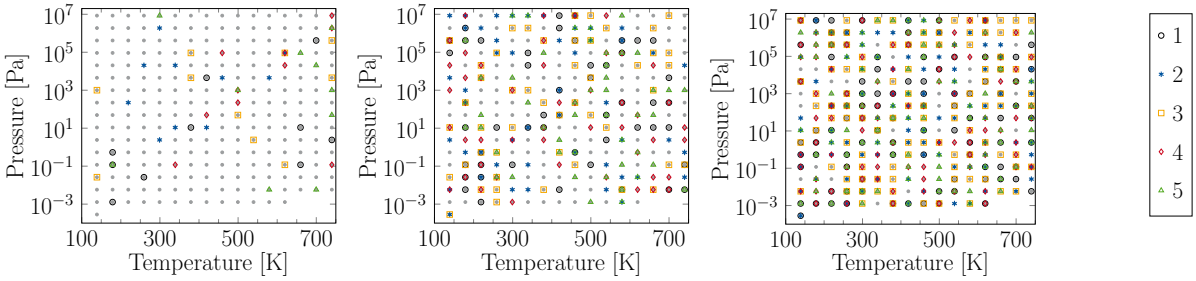
(H1) Both the final upper and lower bound do not change drastically when reducing the size of the dataset.

(H2) The reduction of the dataset allows for a faster global solution of the parameter estimation problem.

Hypothesis (H1) comprises detecting (*i*) *good regions* containing parameter tuples giving low objective values and (*ii*) *bad regions* containing only parameter tuples yielding high objective values. While (*i*) avoids cutting off candidates for the global optimum of the original problem, (*ii*) aims at cutting off unfavorable parts of the B&B tree early based on reduced datasets. Even if we can efficiently prune the B&B tree based on a reduced dataset, i.e., if Hypothesis (H1) is true, we can only save CPU time by data reduction if a smaller dataset implies a smaller computational effort, see Hypothesis (H2).

3.3.1. Setup for studying the global optimization of reduced problems

For this study, we use parameter estimation problem EOS262 described in Section D.3.2.1 as the *base scenario*. We refer to the parameter estimation problems derived by reducing dataset with *sample scenarios*. In detail, we randomly pick 15 data samples from the full dataset $\mathcal{D}$ with $|\mathcal{D}| = 262$. In Scenarios 100_1 to 100_5 we fit the parameters based on 100 data points, in Scenarios 50_1 to 50_5 based on 50 data points, and in Scenarios 10_1 to 10_5 based on 10 data points, see Figure 3.1.
For investigating the impact of data reduction on the optimal solution, we study DGO results over the whole feasible set of the 9 unknown parameters and 100 different subregions of the feasible domain. When dividing the feasible region into subregions, we use closed intervals for the continuous-valued unknown parameters and, in analogy, subsequent values for the discrete unknown parameters. Thus, we call subregions *parameter boxes*.
The different parameter boxes represent different nodes of a potential B&B tree. To consider both good and bad regions, we identified Boxes 1, 2, and 3 which contain parameter values resulting in up to high quality fits (good region), up to medium quality fits, and only low quality fits (bad region), respectively, by function evaluations. To also cover



(a) Scenario 10_1-5 **(b)** Scenario 50_1-5 **(c)** Scenario 100_1-5

Figure 3.1.: Pressure-temperature-pairs used as measurement data for problem EOS262 in the sample scenarios (see legend) and base scenario ($\cdot$). Note that the y-axis is logarithmic.

larger regions, Boxes 4 to 50 are systematic combinations of subregions of the feasible set of different unknown parameters, e.g., we use bounds $c_{10} \in \{0\}$ in Boxes 4 and 5 as well as $c_{10} \in \{1\}$ in Boxes 6 and 7 in combination with bounds $l_{10} \in \{1\}$ in Boxes 4 and 6 as well as $l_{10} \in \{2\}$ in Boxes 5 and 7. For the exact bounds of all parameters, please refer to the supplementary material of [173]. Since we cannot foresee the actual B&B tree, we added Boxes 51 to 100 which contain randomly generated parameter bounds. For simplification of notation, we denote the whole feasible parameter domain with *Box 0*.
By computing lower and upper bounds for the different scenarios and boxes via the standard spatial B&B algorithm, which are as tight as possible given our CPU time limit, we aim at an informative representation of possible nodes of a B&B tree and the corresponding pruning behavior of the B&B algorithm. We do not investigate the optimal parameter values of the different boxes as these do not affect the pruning behavior.
For the optimization, we use our open-source DGO solver MAiNGO version 0.5.0.1 with an absolute and relative optimality tolerance of 10^{-2} and a CPU time limit of 2 h. In preprocessing, we run 100 local searches with LBFGS [129, 148] implemented in the NLOPT toolbox v2.5.0 [103]. MAiNGO invokes CLP version 1.17.0 [69] for solving the lower bounding problem, and LBFGS with up to 10 iterations for solving the upper bounding problem. The relaxations are obtained from MC++ [43]. We perform all calculations on an Intel(R) Core(TM) i5-3570 processor with 3.4 GHz.

3.3.2. Results and discussion

Before running DGO, we evaluated the objective function for all 15 sample scenarios at 3 different points which resulted in a high, medium, and low objective value for the base scenario, see Table B.1 in the Appendix for the respective parameter points. When using too few data points, namely Scenarios 10_1 to 10_5 in this case study, the lower bound for the sample scenarios may be orders of magnitude smaller than the objective value of the base scenario, see Figure 3.2. Since the sample scenarios with 50 and 100 data points result in objective values in a similar order of magnitude than the objective value of the base scenario, we run DGO only for Scenarios 50_1 to 50_5 and 100_1 to 100_5. Hereby, we use the last results reported rather than the final results for the base scenario in Boxes 10, 12, and 13, for Scenario 50_3 in Boxes 28 and 29, for Scenario 50_4 in Boxes 14 and 30, Scenario 50_5 in Boxes 32 and 33 as well as Scenario 100_4 in Box 29 as the DGO within MAiNGO aborts for these 10 out of 1111 cases given by optimizing 11 scenarios over 101

Table 3.1.: Order of magnitude of final lower and upper bound for the base scenario and sample scenarios 50_1 to 50_5 and 100_1 to 100_5 indicated by the minimum value, maximum value, and median when dividing the parameter boxes into two categories

	Base scenario			Sample scenarios		
	min	max	median	min	max	median
Final lower bound						
good regions[a]	$6.01 \cdot 10^{-16}$	$9.84 \cdot 10^{-2}$	$3.14 \cdot 10^{-2}$	0.00	$3.31 \cdot 10^{-1}$	$8.36 \cdot 10^{-3}$
bad regions[b]	$4.35 \cdot 10^{4}$	$1.41 \cdot 10^{17}$	$2.17 \cdot 10^{8}$	$1.64 \cdot 10^{-17}$	$6.57 \cdot 10^{15}$	$2.61 \cdot 10^{5}$
Final upper bound						
good regions[a]	$1.55 \cdot 10^{-5}$	$6.50 \cdot 10^{4}$	$4.91 \cdot 10^{1}$	$6.36 \cdot 10^{-17}$	$4.98 \cdot 10^{5}$	$9.24 \cdot 10^{-2}$
bad regions[b]	$4.51 \cdot 10^{4}$	$1.41 \cdot 10^{17}$	$2.17 \cdot 10^{8}$	$6.96 \cdot 10^{-3}$	$6.57 \cdot 10^{15}$	$2.61 \cdot 10^{5}$

[a]Set of all boxes with lower bound < 0.1 for the base scenario, i.e., boxes which potentially contain a perfect fit

[b]Set of all boxes with lower bound ≥ 0.1 for the base scenario, i.e., boxes which cannot contain a perfect fit

boxes. In particular, the abortion is caused by the third-party solver CLP used for solving the lower bounding problem.

At first, we focus on Hypothesis (H1). Let us recall that we aim for an algorithm using a lower bound calculated based on a reduced dataset for solving the original problem. Transferred to the study on reduced problems presented in this section, this means to compare the lower bound of a sample scenario with the upper bound of the base scenario for deciding whether to prune the respective node. Since we want to investigate whether the data reduction changes the pruning behavior drastically, we are interested in the order of magnitude of the bounds on the optimal solution value rather than the exact values.

Overall, the order of magnitude for the final upper and lower bounds is consistent, i.e., close to the optimal solution value 0 or significantly larger, for any optimized scenario within each of the parameter boxes, cf. Figure 3.3 and Figure 3.4. Note that all values smaller than the optimality tolerance $\varepsilon = 0.01$ are treated as perfect fit by the spatial B&B algorithm.

For a more detailed analysis of the order of magnitude of the final bounds, we cluster the boxes into two categories: (i) good regions where the base scenarios may contain a perfect fit according to the final lower bound and (ii) bad regions where the base scenario can only contain low quality fits as proven by a large final lower bound. In the good regions, we obtain small final lower bounds for all sample scenarios, see Table 3.1. In contrast to that, there are sample scenarios indicating a bad region as good, compare the minimum value of $1.64 \cdot 10^{-17}$. However, the median values over the sample scenarios are consistent with the categories, i.e., we have a small final lower bound in good regions and a large final lower bound in bad regions. Similarly, we can find large final upper bound values in good regions, indicating non-converged optimization runs, and – for some sample scenarios – small final upper bounds in bad regions. But again, the median values of the final upper bounds of the sample scenarios are consistent to the categories defined via the base scenario.

In cases with a significant absolute deviation between the lower bounds of a sample scenario and the corresponding base scenario, the sample scenarios always yield smaller final lower bounds. In these cases, the current upper bound of the B&B tree, i.e., the best solution value found so far, may lie between the lower bound of the base scenario and the lower bound of a sample scenario. Hence, data reduction may prevent pruning of such a node and, consequently, lead to a larger B&B tree. To investigate this in more detail, we assume

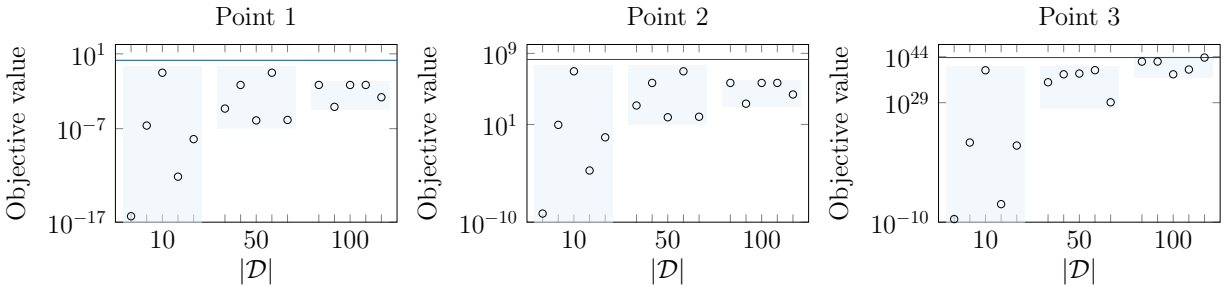


Figure 3.2.: Logarithmic plot of the objective values of problem EOS262 at different parameter points evaluated with the full dataset (–) as well as the reduced datasets of sample scenarios 10_1 to 10_5, 50_1 to 50_5, and 100_1 to 100_5 (o), where the domain reached by evaluating the reduced datasets is shaded ( )

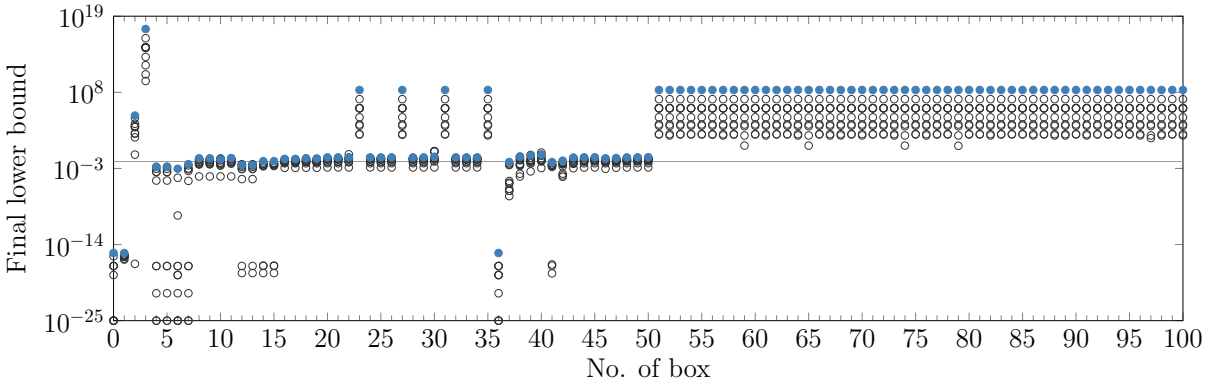


Figure 3.3.: Final lower bound obtained for problem EOS262 with the standard B&B algorithm for the base scenario (•) as well as sample scenarios 50_1 to 50_5 and 100_1 to 100_5 (o) when optimizing over different parameter boxes, where Box 0 is the whole feasible region, with an optimality tolerance of 0.01 (–). Marks on the x-axis indicate values equal to 0.

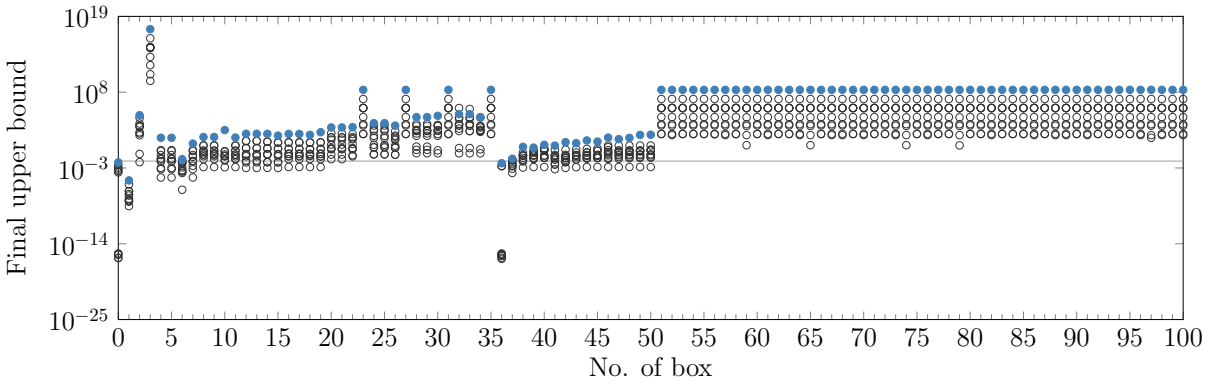


Figure 3.4.: Final upper bound obtained for problem EOS262 with the standard B&B algorithm for the base scenario (•) as well as sample scenarios 50_1 to 50_5 and 100_1 to 100_5 (o) when optimizing over different parameter boxes, where Box 0 is the whole feasible region, with an optimality tolerance of 0.01 (–)

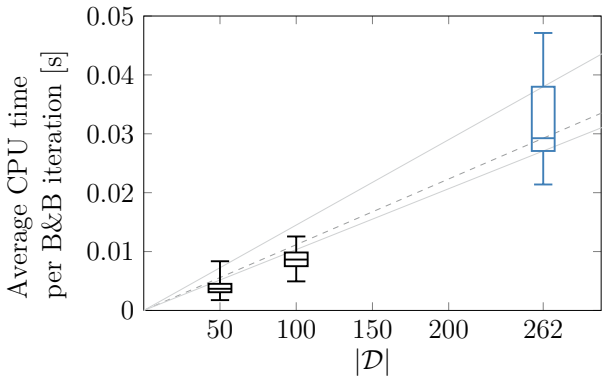


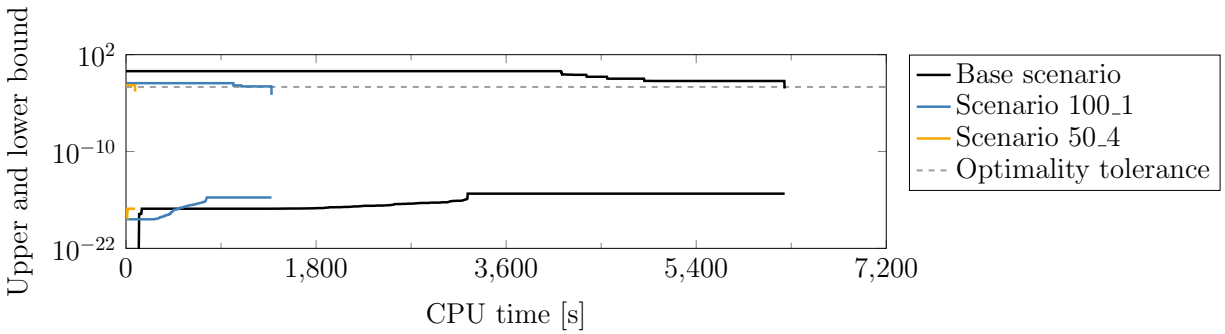
Figure 3.5.: Box-Whisker-plot of the average CPU time per B&B iteration in dependence of the size of the dataset when solving the base scenario (–) as well as sample scenarios 50_1 to 50_5 and 100_1 to 100_5 (–) for problem EOS262 over 100 different parameter boxes and the whole feasible region with auxiliary lines indicating a linear relationship (–, - -)

a B&B tree, where we have the whole feasible range, i.e., Box 0, in the root node and one of the Boxes 1 to 100 in the active child node. In 37 of the 101 boxes, a child node would be pruned when using the base scenario but kept when using (some of) the sample scenarios. Note that these are the cases where the sample scenarios yield small final lower bounds smaller than 0.1 which are larger than the optimality tolerance $\varepsilon = 0.01$, i.e., the values just above the solid line in Figure 3.3. For all other cases, the decision whether to prune does not change when replacing the base scenario with any of the sample scenarios.

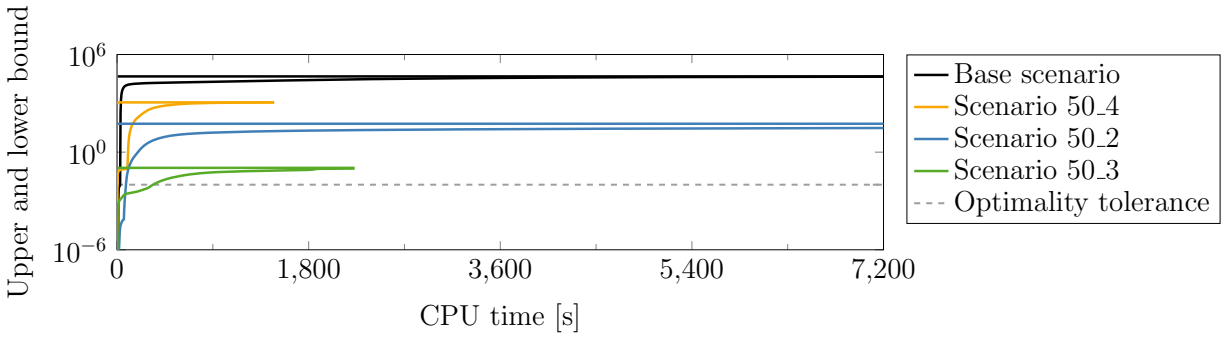
We can prove that the optimal solution value of the sample scenarios is smaller or equal the optimal solution value of the base scenario, compare Lemma 1. As we cannot converge within our CPU time limit of 2 h for some of the cases, see Figure 3.7, there are still some cases where the final lower bound of the sample scenario is larger than the final lower bound of the base scenario. However, this only happens for 16 cases in total with a marginal maximum distance of 0.3 between the lower bounds of the base and the sample scenarios. Note that there are also two cases where we obtain a larger final upper bound for the sample scenario than for the corresponding base scenario within our CPU time limit, namely for Scenario 100_4 in Boxes 32 and 33.

Secondly, we check Hypothesis (H2) regarding the reduction of the CPU time by data reduction. We aim at accelerating the standard B&B algorithm by using a reduced dataset in the root nodes and let it grow until converging to the full dataset. Thus, we investigate the CPU time per B&B iteration rather than the CPU time of the complete DGO, whereby B&B iteration means processing one node including lower and upper bounding as well as applying bound tightening heuristics. Figure 3.5 shows that we can typically expect a CPU time reduction for the B&B iterations when using a reduced dataset. In fact, the scenarios with 100 data points are on average 4 times faster and the scenarios with 50 data points are on average 8 times faster than the base scenario in the respective box. However, single scenarios may be exceptions to this rule. For example, the quickest case with $|\mathcal{D}| = 100$, namely Scenario 100_5 in Box 61, is faster than the slowest 22% cases with $|\mathcal{D}| = 50$.

The results for the average CPU time per iteration cannot be easily transferred to the total CPU time of the DGO as the data reduction changes the optimization problem. The actual speed of convergence depends on the tightness of the lower and upper bounds for the specific



(a) Box 0, i.e., the whole feasible domain



(b) Box 2 which is a "bad region"

Figure 3.6.: Convergence behavior of the base scenario and exemplary sample scenarios over the whole feasible domain (Box 0) and a parameter box which cannot contain the optimal solution (Box 2)

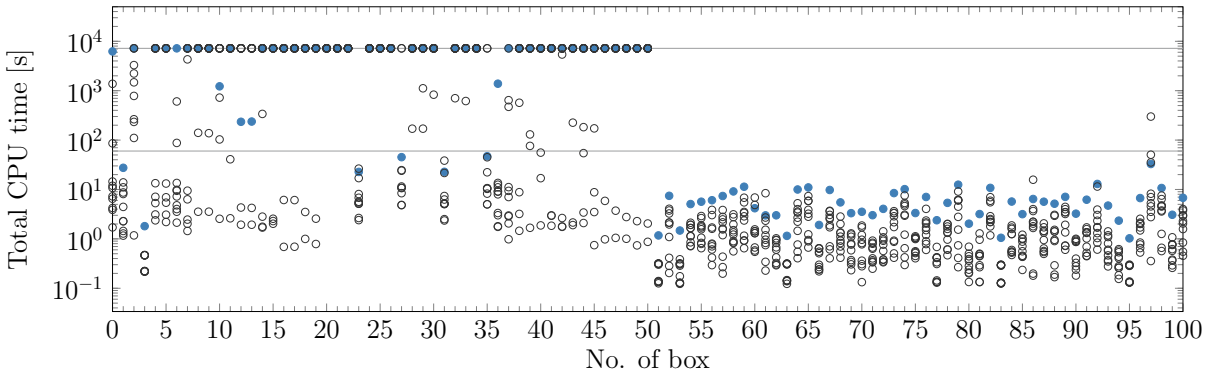


Figure 3.7.: Total CPU time used by the standard B&B algorithm to solve problem EOS262 with the base scenario (•) as well as sample scenarios 50_1 to 50_5 and 100_1 to 100_5 (o) when optimizing over different parameter boxes, where Box 0 is the whole feasible region, with lines (–) indicating the CPU time limit of 2 h (at the top) and 60 s (at the bottom)

Table 3.2.: Computational performance of sample scenarios 50_1 to 50_5 and 100_1 to 100_5 in comparison to the base scenario when running the standard B&B algorithm for problem EOS262 over the whole feasible set and the different parameter boxes

Scenario	Total number	No. scen. with runtime $< 2\,\mathrm{h}$	$< 60\,\mathrm{s}$	$<$ base[a]	r_k	Median of r_k^{CPU}	No. scen. with r_k^{CPU} $< r_k$	$< \frac{1}{2} r_k$
Whole feasible region								
Base	1	1	0	–	1	1	–	–
100_1 to 100_5	5	5	4	5	0.38	0.002	5	5
50_1 to 50_5	5	5	4	5	0.19	0.001	5	4
Boxes 1 to 100								
Base	100	60	56	–	1	1	–	–
100_1 to 100_5	500	301	292	297	0.38	0.44	239	47
50_1 to 50_5	500	380	358	375	0.19	0.12	327	187

[a]CPU time required for optimizing the base scenario over the same parameter box

parameter estimation problem solved which, in turn, depends on the actual solutions found by the local solvers, the choice of the linearization points, and the performance of the bound tightening techniques, cf. Figure 3.6. Nevertheless, we observe that all sample scenarios are faster than the base scenario when optimizing over the whole feasible range, and more than 98% of the converged sample scenarios are faster in Boxes 1 – 100, see Figure 3.7 and Table 3.2. Note that the vast majority of both the base and the sample scenarios converges either within 1 minute or hits the the CPU time limit, namely 64% and 32%, respectively, of all cases. In particular, in Boxes 51 to 100 the standard B&B algorithm converges within the root node for most of the cases.

We are not interested in the DGO of a reduced problem but aim at exploiting reduced datasets for the DGO of the original problem. Thus, we define CPU time ratio

$$r_k^{\mathrm{CPU}} := \frac{\text{CPU time of sample scenario } k}{\text{CPU time of base scenario}}$$

and data ratio

$$r_k := \frac{|\mathcal{D}_k|}{|\mathcal{D}|}$$

for evaluating the time gain for a sample scenario k. In case of equality $r_k^{\mathrm{CPU}} = r_k$, the CPU time increases linearly in the number of data points. If $r_k^{\mathrm{CPU}} \ll r_k$, we can solve comparatively many samples while still reducing the overall CPU time. Table 3.2 shows that the total CPU time for solving any of the sample scenarios over the whole feasible domain is reduced even more than the size of the dataset, i.e., $r_k^{\mathrm{CPU}} < r_k$. The same holds for 83% of the converged sample scenarios in Boxes 1 to 100. For 35% of the converged sample scenarios over all boxes including Box 0, the CPU time ratio is even twice as small as the data ratio.

3.3.3. Conclusion

With the study on reduced problems in this section, we confirmed that

(H1) the order of magnitude of both objective function values and lower bounds is preserved

(H2) the average CPU time per iteration can be expected to decrease

when reducing the size of the dataset.
Regarding Hypothesis (H1), good and bad regions of the base scenario can be distinguished well by sample scenarios with a sufficient number of data points, in particular, $|\mathcal{D}| \in \{50, 100\}$ in this case study. In general, data reduction may allow for lower objective values and, thus, blow up the B&B tree by keeping more nodes than required. However, the order of magnitude of the lower bound in different subregions of the feasible set is identified correctly by the sample scenarios, which already allows for pruning nodes containing only low quality fits. On the other hand, some of the sample scenarios give slightly larger lower bounds than the corresponding base scenario in good regions, which can result in pruning the global solution in the worst case. For an algorithm exploiting reduced datasets, we therefore propose data augmentation when approaching good regions. Note that the reduced dataset should represent the complete dataset adequately as solving a parameter estimation problem often means to balance model prediction errors from different regions. This means, for example, considering measurements in both liquid and gas phase for fitting problem EOS262. Note that our positive results regarding Hypothesis (H1) indicate that the presented samples preserve the properties of the full dataset sufficiently, even though they are chosen randomly.
Hypothesis (H2) is confirmed for the average CPU time per B&B iteration, while the conclusion for the total CPU time of the DGO is more diverse. The DGO of most of the sample scenarios converges faster than DGO of the corresponding base scenario, while the optimization of some sample scenarios takes as long as or even longer than the base scenario in some of the subregions of the feasible set. Note again that we are looking for the optimal solution of the original problem by exploiting reduced datasets rather than for the optimal solution of a reduced problem.
Since (*i*) the sample scenarios can identify good and bad regions and (*ii*) the average CPU time per B&B iteration tends to decrease when using reduced datasets, we expect an acceleration of the B&B algorithm when replacing the full dataset with reduced datasets in some of the B&B nodes. For the problem EOS262 used in this study, the B&B tree could process 8 or 4 times the number of iterations within the same CPU time limit when reducing the dataset from 262 to 50 or 100 data points. Since the solution of complex estimation problems often takes multiple thousand iterations, fathoming the B&B tree may be sped up even if hypotheses (H1) and (H2) fail in a small number of nodes. Moreover, data augmentation eventually closes the gap between sample scenarios and base scenario. Consequently, replacing the base scenario by sample scenarios at the beginning of the B&B algorithm and successively augmenting data until we reach the full dataset at the end appears promising for making the DGO of large-scale parameter estimation problems tractable.

3.4. B&B algorithm with growing datasets

3.4.1. Fathoming B&B nodes based on reduced datasets

For developing valid lower and upper bounding procedures for the B&B nodes based on reduced datasets, we make the following assumption.

Assumption 4.
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.

As the key idea behind the proposed extension of a B&B algorithm, we split the overall error into the error terms of each single data point. Consequently, we add another assumption.

Assumption 5.

(i) 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 $(\boldsymbol{x}_d, y_d) \in \mathcal{D}$.

(ii) 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 $\mathcal{P}$.

For any set $\mathcal{D}_{\text{red}} \subseteq \mathcal{D}$, we define $g^{\text{cv}}_{\mathcal{D}_{\text{red}}}(\boldsymbol{p}) := \sum_{(\boldsymbol{x}_d,y_d)\in\mathcal{D}_{\text{red}}} g^{\text{cv}}(\boldsymbol{p};\boldsymbol{x}_d, y_d)$ and obtain the following Lemma.

Lemma 1. *Let Assumptions 4 and 5 hold. Then,*

(i) $g^{cv}_{\mathcal{D}_{red}}(\boldsymbol{p})$ *is a convex underestimator of both* $g_{\mathcal{D}_{red}}$ *and* $g_{\mathcal{D}}$ *over* $\mathcal{P}$ *for any* $\mathcal{D}_{red} \subseteq \mathcal{D}$,

(ii) a lower bound on the globally optimal objective value of (PE) *is given for any* $\mathcal{D}_{red} \subseteq \mathcal{D}$ *by*

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

(iii) $g_{\mathcal{D}}(\boldsymbol{p})$ *is an upper bound on the globally optimal objective value of* (PE) *for any* $\boldsymbol{p} \in \mathcal{P}$ *satisfying all inequality constraints.*

The proof of Lemma 1 follows from our definitions and from the fact that the sum of convex underestimators is a convex underestimator of the sum for continuous functions. In particular, the nonnegativity of the squared error terms and of the convex underestimators as stated in Assumption 5(i) yields the statement in Lemma 1(i). Note that applying $h^{\text{cv}}(\boldsymbol{x}_d, y_d;\boldsymbol{p}) \le 0$ for any $(\boldsymbol{x}_d, y_d) \in \mathcal{D}_{\text{red}}$ just means to skip some constraints from (PE) if $\mathcal{D}_{\text{red}} \subsetneq \mathcal{D}$, while the data-independent constraints $\widetilde{h}^{\text{cv}}$ remain exactly the same when using a reduced dataset.
With Lemma 1, we gain valid lower bounds on the solution values of (PE) utilizing reduced datasets. Meaning, we can potentially save computational effort for the solution of (PE) by utilizing a smaller estimation problem for the calculations of convex relaxations to obtain

convex problem (LBP), its linearization and the subsequent linear optimization within the lower bounding procedure. Similarly, we can obtain a valid upper based on a smaller problem if we plug in a solution point from a local nonlinear optimization of the reduced problem into the original problem and this point is feasible for the full dataset as well.
For the further analysis of lower and upper bounds, we denote with *full lower bound* l_k^{full} and *reduced lower bound* l_k^{red} the lower bound obtained by optimizing (LBP) over parameter domain $\mathcal{P}_k$ based on the full dataset and a reduced dataset, respectively. If the corresponding *full lower bounding problem* and *reduced lower bounding problem* are feasible, we denote an optimal point by $\boldsymbol{p}_k^{\text{lb,full}}$ and $\boldsymbol{p}_k^{\text{lb,red}}$, respectively.
The downside of the reduced lower bound being valid for the original problem is that it is potentially less tight than the full lower bound. If the gap between reduced and full lower bound prevents pruning, the B&B tree grows larger than necessary which may nullify the time savings due to data reduction. In fact, there are two reasons potentially preventing pruning when using a reduced dataset for lower bounding: (i) the full lower bound would also be smaller than the current upper bound, or (ii) the full lower bound would be larger than the current upper bound but the approximate reduced lower bound is not. In case (i), we want to branch the node to narrow down the region containing the optimal solution. In case (ii), we want to augment the dataset to decrease the gap between reduced and full lower bound. However, we only have information on the reduced lower bound at the beginning of the proposed B&B algorithm and, thus, do not know which case we are in. For the decision whether to branch or augment, we therefore introduce *augmentation rules*.

3.4.2. Augmentation rules

Subroutine 1 Augment or branch at iteration k

Input: Set of active nodes $\mathcal{N}_k$, currently processed node $N_k = (\mathcal{P}_k, \mathcal{D}_k)$, augmentation rule $\mathbb{A}$, augmentation size φ, full dataset $\mathcal{D}$
Output: Updated sets $\mathcal{N}_{k+1}$ and $\mathcal{N}_{k+1}^{\text{f}}$
if $\mathbb{A}(N_k) =$True **then**
  Choose $\mathcal{D}_{\text{new}} \subseteq \mathcal{D} \setminus \mathcal{D}_k$ such that $|\mathcal{D}_{\text{new}}| = \min\{\text{rd}\,(\varphi \cdot |\mathcal{D}|)\ ,\ |\mathcal{D} \setminus \mathcal{D}_k|\}$;
  $N^{\text{new}} := (\mathcal{P}_k, \mathcal{D}_k \cup \mathcal{D}_{\text{new}})$;
  $\mathcal{N}_{k+1} := (\mathcal{N}_k \setminus N_k) \cup N^{\text{new}}$;
else
  Branch N_k by partitioning $\mathcal{P}_k = \mathcal{P}_{k,1} \cup \mathcal{P}_{k,2} \cup \dots$;
  $N^{\text{new},1} := (\mathcal{P}_{k,1}, \mathcal{D}_k)$, $N^{\text{new},2} := (\mathcal{P}_{k,2}, \mathcal{D}_k)$, $\dots$;
  $\mathcal{N}_{k+1} := (\mathcal{N}_k \setminus N_k) \cup N^{\text{new},1} \cup N^{\text{new},2} \cup \dots$;
end if

We associate all nodes of the B&B tree with a parameter domain and a dataset. An augmentation rule $\mathbb{A} : \mathcal{N} \to \{\text{True}, \text{False}\}$ decides in each B&B iteration k for the processed node $N_k = (\mathcal{P}_k, \mathcal{D}_k) \in \mathcal{N}$ whether to branch parameter domain $\mathcal{P}_k \subseteq \mathcal{P}$, which means adding child nodes whose parameter domains form a partition of domain $\mathcal{P}_k$ [see, e.g., 22, 97, 131, 195], or whether to augment dataset $\mathcal{D}_k \subseteq \mathcal{D}$, see Subroutine 1. Note that the child node inherits the parameter domain $\mathcal{P}_k$ of its parent node when augmenting. For the time being, we pick the additional data points $\mathcal{D}_{\text{new}}$ randomly from $\mathcal{D} \setminus \mathcal{D}_k$. The number of data points to be added is determined by an a-priori given *augmentation size* $\varphi \in (0, 1]$

which is a fraction of the size of the full dataset. For example, with $|\mathcal{D}| = 100$ and $\varphi = 0.25$ we would add 25% of the full dataset, i.e., 25 new data points, when augmenting. With an initial dataset containing 10% of the available data points, the augmentation procedure would determine four different reduced datasets consisting of 10, 35, 60, and 85 data points. For the proof of convergence, we use augmentation rules which eventually lead to the full dataset, i.e., which *complete finitely*.

Definition 1. 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 3.8.
The augmentation rule $\mathbb{A} : \mathcal{N} \to \{\text{True}, \text{False}\}$ *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 $\exists J < \infty : \mathcal{D}_{k_j} = \mathcal{D}\ \forall j \geq J$.

Note that we have $\mathcal{P}_{k_{j_2}} \subseteq \mathcal{P}_{k_{j_1}}$ and $\mathcal{D}_{k_{j_2}} \supseteq \mathcal{D}_{k_{j_1}}$ for any two nodes $N_{k_{j_1}}, N_{k_{j_2}}$ with $j_2 > j_1$ in an infinite nested sequence of nodes $\{N_{k_j}\}_{j\to\infty}$ since we either branch the parameter domain and keep the dataset or augment the dataset and keep the parameter domain in each iteration of the B&B algorithm.

3.4.2.1. CONST: Augmenting with fixed frequency

An intuitive augmentation rule is to augment every c^{th} iteration of the B&B algorithm with growing datasets, where $c \in \mathbb{N}$ is an a-priori given constant. However, depending on the node selection strategy, such a rule would pick nodes at different depths within the B&B tree for augmenting. In the absence of additional information, we therefore augment if the depth of the current node is a multiple of constant c instead. We obtain an augmentation rule which completes finitely.

Proposition 1. *Let an initial dataset $\mathcal{D}_0 \subseteq \mathcal{D}$ be given. If the augmentation step defined in Subroutine 1 is performed with a constant augmentation size $\varphi \in (0, 1]$, then augmentation rule*

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

with an a-priori known constant $c \in \mathbb{Z}$ completes finitely.

Proof. Let $\{N_{k_j}\}_{j\to\infty}$ with $N_{k_j} = (\mathcal{P}_{k_j}, \mathcal{D}_{k_j})$ be a sequence of nested nodes generated by the B&B algorithm with growing datasets. By definition, we have $\text{depth}(N_{k_j}) = j$ for any sequence of nested nodes starting with the root node.
We need $l := \lceil \frac{|\mathcal{D}| - |\mathcal{D}_0|}{\lceil \varphi \cdot |\mathcal{D}| \rceil} \rceil$ subsequent augmentations to reach the full dataset. According to augmentation rule $\mathbb{A}_{\text{CONST}}$, augmentation is triggered every c^{th} depth. Thus, we obtain the full dataset in nodes with depth $l \cdot c$, i.e., for any $j \geq J$ with $J := l \cdot c$. □

Since augmentation rule CONST completes finitely, see Proposition 1, we can prove the convergence of the B&B algorithm with growing datasets in finite time, cf. Theorem 2. However, we cannot make statements about the speed of convergence. As an example, see the results in Table 3.3 obtained for problem EOS262, see Section D.3.2.1 for the model formulation, with the numerical setup explained in Section 3.6.1. While we can almost half the CPU time required for the DGO with the standard B&B algorithm when using augmentation rule CONST with frequency $c = 5$ within the B&B algorithm with growing

Table 3.3.: Total CPU time and number of nodes processed for convergence of the standard B&B algorithm as well as the B&B algorithm with growing datasets applying augmentation rule CONST with different frequencies c when solving the model EOS262

	Total CPU time	No. of nodes processed
Standard B&B	111.1 min	110 901
CONST $c = 5$	126.8 min	113 897
CONST $c = 10$	62.1 min	129 732

datasets, the CPU time increases with frequency $c = 10$. Note that the B&B algorithm with growing datasets processes more nodes in both cases: If frequency c is chosen too small, see results with $c = 5$, we reach the full dataset before we could prune any node of the B&B tree based on a reduced dataset. In that way, growing datasets just increase the size of the B&B tree and, consequently, the total CPU time. If frequency c is chosen sufficiently large, see results with $c = 10$, we can approach the global optimum faster by processing more (reduced) nodes in the same time. As we can hardly make a-priori statements on the right choice of frequency c for general parameter estimation problems, we propose further augmentation rules which make use of an approximation of the full lower bound for the decision whether to branch or augment.

3.4.2.2. SCALING: Augmenting with stochastic guarantees

Augmentation rule SCALING aims to branch if we could have pruned the node when using the full lower bound. As we do not know the full lower bound, we estimate the full lower bound by scaling the reduced lower bound. In particular, we assume in this section that a solution of the lower bounding problem (LBP) exists, i.e., that there is a feasible point in this node. Otherwise, the node is pruned without augmenting or branching.
As indicated by our results in Section 3.3 and Appendix B.2, we assume that the mean squared error does not change significantly when changing the dataset

$$\frac{1}{|\mathcal{D}_k|} \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}_k} g^{\mathrm{cv}}(\boldsymbol{p}; \boldsymbol{x}_d, y_d) \approx \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} g^{\mathrm{cv}}(\boldsymbol{p}; \boldsymbol{x}_d, y_d) \tag{3.1}$$

or fulfills at least a sublinear relation

$$\frac{1}{|\mathcal{D}_k|} \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}_k} g^{\mathrm{cv}}(\boldsymbol{p}; \boldsymbol{x}_d, y_d) \leq \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} g^{\mathrm{cv}}(\boldsymbol{p}; \boldsymbol{x}_d, y_d) \ . \tag{3.2}$$

Evaluating (3.1) at the optimal point $\boldsymbol{p}_k^{\mathrm{lb,red}}$ gives the *scaled lower bound*

$$\widehat{l}_k^{\mathrm{scaled}} := \frac{|\mathcal{D}|}{|\mathcal{D}_k|} \cdot l_k^{\mathrm{red}}$$

as an approximation of the full lower bound. We obtain the augmentation rule

$$\mathbb{A}_{\mathrm{SCALING}^\star}(N_k) := \begin{cases} \text{True}, & \text{if } \widehat{l}_k^{\mathrm{scaled}} \geq u_k - \varepsilon \\ \text{False}, & \text{else} \end{cases} \ . \tag{3.3}$$

Note that both assumptions (3.1) and (3.2) may be violated, particularly in models sensitive to single data points. Consequently, $\widehat{l}_k^{\text{scaled}}$ is a *heuristic lower bound* which may not be valid for the original problem. In the following, we denote heuristic lower bounds with $\widehat{l}_k$ compared to valid lower bounds l_k.
For a fixed dataset $\mathcal{D}$, see Assumption 6, we can show that augmenting via SCALING is most likely triggered only if we could prune based on the full dataset. Note that dataset $\mathcal{D}$ may still contain deviations from the model predictions with the true model parameters. However, in this thesis, we do not account for any random behavior in the data.

Assumption 6.

(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 $\mathcal{D}_k \subsetneq \mathcal{D}$ be a reduced dataset with an a-priori fixed size, where the data points in $\mathcal{D}_k$ are picked randomly from $\mathcal{D}$ such that $\mathcal{D}_k$ follows a discrete uniform probability distribution over all subsets of $\mathcal{D}$ with size $|\mathcal{D}_k|$.

Lemma 2. *If $\mathcal{D}_k \subsetneq \mathcal{D}$ is chosen according to Assumption 6, then*

$$\text{EV}\left[l_k^{red}\right] \leq \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \cdot l_k^{full}.$$

Proof. The definition of the reduced lower bound, the monotonicity of the expected value, and the optimality of $\boldsymbol{p}_k^{\text{lb,red}}$ yield $\text{EV}\left[l_k^{\text{red}}\right] = \text{EV}\left[g_{\mathcal{D}_k}^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,red}})\right] \leq \text{EV}\left[g_{\mathcal{D}_k}^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,full}})\right]$. From the discrete uniform distribution assumption, it follows for any single data point $(\boldsymbol{x}_d, y_d) \in \mathcal{D}$ that $\mathbb{P}((\boldsymbol{x}_d, y_d) \in \mathcal{D}_k) = \frac{|\mathcal{D}_k|}{|\mathcal{D}|}$. In combination with the linearity of expectation, we obtain $\text{EV}\left[g_{\mathcal{D}_k}^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,full}})\right] = \frac{|\mathcal{D}_k|}{|\mathcal{D}|} g_{\mathcal{D}}^{\text{cv}}(\boldsymbol{p}_k^{\text{lb,full}}) = \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \cdot l_k^{\text{full}}$. □

Theorem 1. *Let Assumptions 4 and 5 hold and $\tilde{\varepsilon} > 0$ be given. Let $\mathcal{D}_k \subsetneq \mathcal{D}$ be chosen according to Assumptions 6. Assume augmentation rule*

$$\mathbb{A}_{\text{SCALING}}(N) := \begin{cases} \textit{True}, & \textit{if } \rho \cdot \widehat{l}_k^{scaled} \geq u_k - \tilde{\varepsilon} \\ \textit{False}, & \textit{else} \end{cases} \tag{3.4}$$

with a constant $0 < \rho \leq 1$.
If $\tilde{\varepsilon} < u_k$ and $l_k^{full} < u_k - \tilde{\varepsilon}$, then the probability for augmenting is less than ρ.

Proof. Due to the nonnegativity of g_k^{cv} and $\tilde{\varepsilon} < u_k \Leftrightarrow u_k - \tilde{\varepsilon} > 0$, we can apply Markov's inequality

$$\begin{aligned} \mathbb{P}\left(\rho \cdot \widehat{l}_k^{\text{scaled}} \geq u_k - \tilde{\varepsilon}\right) &= \mathbb{P}\left(\rho \cdot \frac{|\mathcal{D}|}{|\mathcal{D}_k|} \cdot l_k^{\text{red}} \geq u_k - \tilde{\varepsilon}\right) = \mathbb{P}\left(l_k^{\text{red}} \geq \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \cdot \frac{1}{\rho} \cdot (u_k - \tilde{\varepsilon})\right) \\ &\leq \frac{\text{EV}\left[l_k^{\text{red}}\right]}{\frac{|\mathcal{D}_k|}{|\mathcal{D}|}\frac{1}{\rho}(u_k - \tilde{\varepsilon})}. \end{aligned}$$

With Lemma 2 and $0 \leq l_k^{\text{full}} < u_k - \tilde{\varepsilon}$, the statement follows. □

Table 3.4.: Datasets as well as corresponding optimal and scaled optimal objective value for the estimation problem described in Example 1

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

Theorem 1 gives us a theoretical bound on the probability that $\mathbb{A}_{\text{SCALING}}$ triggered augmenting although we could not prune the node based on the full lower bound. Note that we introduced an *augmentation weight* ρ into $\mathbb{A}_{\text{SCALING}^\star}$ (3.3) to obtain $\mathbb{A}_{\text{SCALING}}$ (3.4) such that we can investigate the numerical behavior based on different bounds on the probability. Note further that condition $\tilde{\varepsilon} < u_k$ is a weak assumption, since $\tilde{\varepsilon} \geq u_k$ implies that the B&B algorithm has already converged if $\tilde{\varepsilon}$ equals optimality tolerance ε.
We expect that augmentation rule $\mathbb{A}_{\text{SCALING}}$ is true sufficiently often to reach the full dataset. However, there are pathological cases where this augmentation rule does not complete finitely.

Example 1 (Conflicting data prevents convergence). We want to solve

$$\min_{p \in [0,25]} \sum_{(x_d, y_d) \in \mathcal{D}} (p \cdot x_d - y_d)^2$$

with full dataset $\mathcal{D} = \{(1, 0), (1, 0.6), (1, 1)\}$, where we find a linear function $f(x; p) = p \cdot x$ through the origin and three data points at the same input $x = 1$, with a naive implementation of the B&B algorithm with growing datasets using augmentation rule $\mathbb{A}_{\text{SCALING}^\star}$, i.e., $\mathbb{A}_{\text{SCALING}}$ with $\rho = 1$. Assume that the reduced datasets are chosen as in Table 3.4. Even if the lower bounds l_0 and l_1 calculated based on reduced datasets $\mathcal{D}_0$ and $\mathcal{D}_1$, respectively, are exact, i.e., equal the respective optimal objective, gaps would remain $l_0 \ll l_1 \ll l^{\text{full}} \leq u$. Branching does not help either since we already assume exact lower bounds. With that, neither augmenting nor convergence is possible. □

While SCALING has favorable statistical properties, cf. Theorem 1, we cannot make deterministic statements on the tightness of the approximation $\widehat{l}_k^{\text{scaled}} \approx l_k^{\text{full}}$ for general models. Similarly, the convergence behavior of the reduced lower bounds to their expected value, namely the full lower bound, see Lemma 2, or the actual distance between reduced and full lower bounds depends highly on the specific model equations and datasets used. Instead of investigating such statements for specific models, we derive statements for general estimation problems (PE) based on out-of-sample estimation in the following.

3.4.2.3. COMBI: Augmenting based on out-of-sample estimation

Taking up on the motivation for augmentation rule SCALING, we want to augment the dataset whenever the gap between reduced and full lower bound prevented pruning. To obtain a better approximation of the full lower bound based on reduced datasets, we propose to apply out-of-sample evaluation. 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 that the out-of-sample lower bound is 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 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

$$\min_{p \in [0,10]} \sum_{(x_d, y_d) \in \mathcal{D}} (p \cdot x_d - y_d)^2 \quad . \tag{3.5}$$

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 (3.5), 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}. □

Reduced lower bound l_k^{red} and out-of-sample lower bound $\widehat{l}_k^{\text{oos}}$ are summations of nonnegative addends which are given as function evaluations at the same point $\boldsymbol{p}_k^{\text{lb,red}}$ in parameter domain $\mathcal{P}_k$. Thus, we expect the reduced lower bound being smaller for small reduced datasets $\mathcal{D}_k$, meaning at the beginning of the B&B algorithm, and the out-of-sample lower bound being smaller for small $\mathcal{D} \setminus \mathcal{D}_k$, meaning towards the end of the B&B algorithm.

Postulate 2.

(i) For most B&B nodes with $|\mathcal{D}_k| \ll |\mathcal{D}|$, we have $l_k^{red} \leq \widehat{l}_k^{oos}$.

(ii) For most B&B nodes with $|\mathcal{D}_k| \approx |\mathcal{D}|$, we have $\widehat{l}_k^{oos} \leq l_k^{red}$.

Postulate 2 is confirmed by numerical experiments with the real-world estimation problems described in Appendix D. The quantification of *most nodes* depends on the actual distribution of the reduced and the out-of-sample lower bound in dependence of the randomly chosen reduced dataset and, hence, highly on the actual model function f as well as the actual full dataset $\mathcal{D}$. Again, pathological cases may violate Postulate 2.
With the help of both reduced and out-of-sample lower bound, we can enclose the full lower bound.

Proposition 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

$$\begin{aligned}
&\sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,full}}; \boldsymbol{x}_d, y_d) \\
&\quad = \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} \Big(g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,red}}; \boldsymbol{x}_d, y_d) + g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,full}}; \boldsymbol{x}_d, y_d) - g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,red}}; \boldsymbol{x}_d, y_d) \Big) \\
&\quad = \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}_k} g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,red}}; \boldsymbol{x}_d, y_d) + \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D} \setminus \mathcal{D}_k} g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,red}}; \boldsymbol{x}_d, y_d) \\
&\qquad + \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} \Big(g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,full}}; \boldsymbol{x}_d, y_d) - g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,red}}; \boldsymbol{x}_d, y_d) \Big) \ . \qquad (3.6)
\end{aligned}$$

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

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

and, together with Lipschitz continuity,

$$\begin{aligned}
&- \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} \Big(g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,full}}; \boldsymbol{x}_d, y_d) - g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,red}}; \boldsymbol{x}_d, y_d) \Big) \\
&\qquad = \Big| \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} \Big(g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,full}}; \boldsymbol{x}_d, y_d) - g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,red}}; \boldsymbol{x}_d, y_d) \Big) \Big| \\
&\qquad \overset{\text{additivity}}{\leq} \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} \left| g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,full}}; \boldsymbol{x}_d, y_d) - g^{\mathrm{cv}}(\boldsymbol{p}_k^{\mathrm{lb,red}}; \boldsymbol{x}_d, y_d) \right| \\
&\qquad \overset{\text{Lipschitz}}{\leq} \sum_{(\boldsymbol{x}_d, y_d) \in \mathcal{D}} L_d \cdot ||\boldsymbol{p}_k^{\mathrm{lb,full}} - \boldsymbol{p}_k^{\mathrm{lb,red}}||. \qquad (3.7)
\end{aligned}$$

Inserting (3.7) in (3.6) gives

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

which is the same as the first inequality of Proposition 2.
For the second inequality, we observe

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

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

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

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 Proposition 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 estimation problems explained in Appendix D. Consequently, we expect numerical advantages when replacing the full lower bound with the combined lower bound as formalized in the following postulates.

Postulate 3.

(i) For most B&B nodes, the difference between combined lower bound $\widehat{l}_k^{combi}$ and full lower bound l_k^{full} is vanishingly small.

(ii) 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}.

Based on Proposition 2, we assume that the combined lower bound gives a tighter approximation of the full lower bound than the scaled lower bound. Thus, we propose augmentation rule COMBI

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

3.4.2.4. TOL: Augmenting instead of tight pruning

Finally, we recall that the motivation for the augmentation rule was to trigger augmentation whenever we are not sure whether to prune or branch a node based on reduced datasets, compare the end of Section 3.4.1. Thus, we propose to trigger augmenting in case of tight pruning decisions. For this, we introduce an interval of uncertainty around the lower bound l_k used for pruning, and augment if we fall within this tolerance $\widehat{\varepsilon}$ giving augmentation rule TOL

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

In MAiNGO, we apply optimality tolerance ε also to the pruning decision. To ensure a non-empty interval $[\,l_k+\varepsilon, l_k+\widehat{\varepsilon}\,]$ for triggering augmentation, we use ten times the default optimality tolerance for the augmentation tolerance, namely $\widehat{\varepsilon}=0.1$ by default.

3.4.2.5. Hybridization with augmentation rule CONST

We can transform any of the augmentation rules $\mathbb{A}_{\text{AR}}$, AR$\in\{$SCALING,COMBI,TOL$\}$ into a finitely completing augmentation rule by combining it with augmentation rule CONST

$$\mathbb{A}_{\text{HYBRID}}(N) := \big(\mathbb{A}_{\text{AR}}(N) \vee \mathbb{A}_{\text{CONST}}(N)\big) \quad \text{for any } N \in \mathcal{N}\,,$$

compare Proposition 1. Note that frequency c of augmentation rule CONST should be sufficiently large to not completely overrule the other augmentation rule.

3.4.3. Proof of convergence

Algorithm 2 B&B algorithm with growing datasets

Input: Model formulation including parameter domain $\mathcal{P}$ and full dataset $\mathcal{D}$, optimality tolerance ε

Output: Global optimal solution including bounds on optimal objective value

1: Initialize trivial bounds $u_0 := \infty$ and $l := -\infty$ as well as counter $k := 0$, root node $N_0 := (\mathcal{P}, \mathcal{D})$ and set of active nodes $\mathcal{N}_0 := \{N_0\}$;
2: Pre-processing of root node N_0;
3: Pick initial dataset $\mathcal{D}_0 \subsetneq \mathcal{D}$ and reset dataset of root node $N_0 := (\mathcal{P}, \mathcal{D}_0)$;
4: **while** $u_k - l > \varepsilon$ **do**
5: Select next node $N_k \in \mathcal{N}_k$; \\Dataset of active node N_k: $\mathcal{D}_k \subseteq \mathcal{D}$
6: Solve lower bounding problem in N_k;
7: Solve upper bounding problem in N_k and obtain solution point $\boldsymbol{p}^{ub}_{\mathcal{D}_k}$;
8: Evaluate model at $\boldsymbol{p}^{ub}_{\mathcal{D}_k}$ based on $\mathcal{D}$ and obtain upper bound;
9: Update best upper bound found so far u_k;
10: **if** N_k cannot be pruned **then**
11: Call Subroutine 1;
12: **else**
13: $\mathcal{N}_{k+1} := \mathcal{N}_k \setminus N_k$;
14: **end if**
15: $\mathcal{N}^{\text{f}}_{k+1} := \mathcal{N}^{\text{f}}_k \cup N_k$;
16: Update lower bound over all active nodes l;
17: $k := k+1$;
18: **end while**

Based on the findings of Section 3.4.1 for fathoming nodes based on reduced datasets, we propose to extend the standard B&B algorithm by growing datasets as shown in Figure 3.8. The preprocessing in the root node, e.g., bound tightening heuristics and multistart of local searches, is still based on the full dataset, while we start with a reduced dataset for lower bounding, compare also Algorithm 2. To get a valid upper bound for (PE), we compute a local optimum based on the reduced dataset and evaluate the corresponding solution

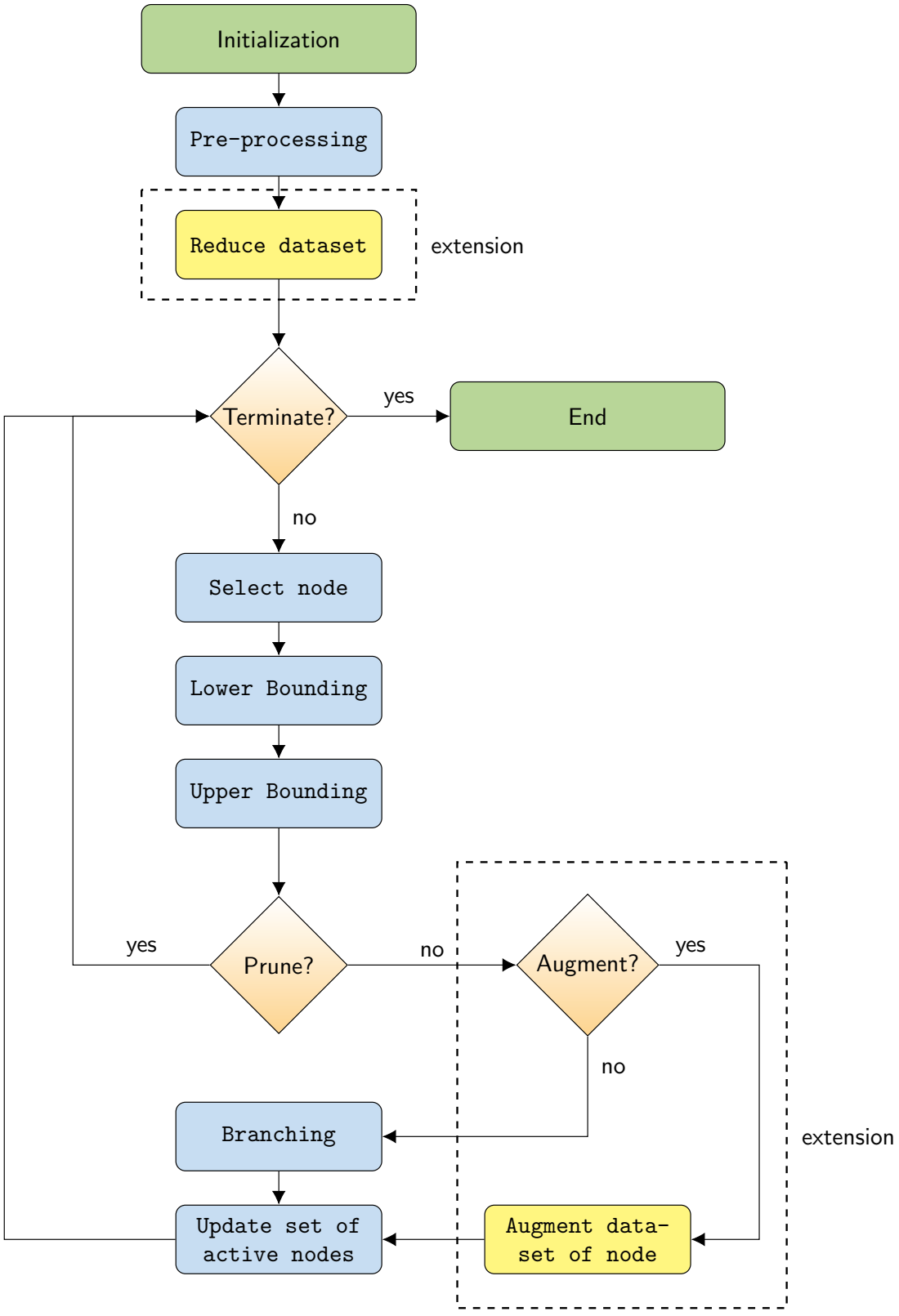


Figure 3.8.: Flow chart of the proposed algorithm, where the dashed boxes highlight the extensions of a standard B&B algorithm to manage growing datasets

point based on the full dataset. Whether to augment or branch is determined based on the augmentation rules presented in Section 3.4.2. Note that both child nodes inherit the dataset from the parent node in branching, cf. Subroutine 1. As a first step, we propose a very simple setup for determining and augmenting the reduced datasets: the size of the initial dataset and the augmentation size φ are user settings, namely 10% and 25% of the full dataset, respectively, by default. The indices of the data points to be chosen are picked randomly for both the initialization and augmentation.
The B&B algorithm with growing datasets is a *deterministic* global optimization algorithm for solving Problem (PE) based on the full dataset, since we obtain valid lower and upper bounds, see Lemma 1. Note that any heuristic lower bounds are only used within augmentation rules, i.e., for the decision whether to augment or branch, while we only use the valid lower bound l_k^{red} for pruning. In fact, we can show that extending a convergent B&B algorithm by growing datasets preserves the convergence to the global optimum of the original problem formulation (PE). For this, we adapt the proof of convergence for a B&B algorithm subject to optimality and feasibility tolerances given in Theorem 5.26 of Locatelli and Schoen (2013) [131].
In detail, the proposed B&B algorithm with growing datasets introduces a dependency on the dataset for the lower and upper bounds. At first, we verify the crucial assumptions of Theorem 5.26 [131]. See Appendix C for the definitions of *exactness in the limit* [131, Def. 5.4], the *isotonic property* [131, Eq. (5.24)], and *exhaustiveness* [131, Def. 5.5] in the notation of this thesis.

Lemma 3. *Let Assumptions 4 and 5 hold. We apply a spatial B&B algorithm with optimality tolerance $\varepsilon > 0$ to (PE). Let the convex underestimator $g_{\mathcal{D}}^{cv}$ be exact in the limit. Let $g^{cv}(\cdot; \boldsymbol{x}_d, y_d)$ satisfy the isotonic property for any $(\boldsymbol{x}_d, y_d) \in \mathcal{D}$. Let the subdivision process of the B&B algorithm be exhaustive.*
If we use the B&B algorithm with growing datasets depicted in Figure 3.8 with iteration index k, optimality tolerance $\varepsilon > 0$, and an augmentation rule $\mathbb{A}$ which completes finitely, then

(i) each element of the generated sequence $(g_{\mathcal{D}_k}^{cv})_k$ satisfies the isotonic property

(ii) each element of $(g_{\mathcal{D}_k}^{cv})_k$ is a convex underestimator of $g_{\mathcal{D}}$

(iii) each element of $(g_{\mathcal{D}_k}^{cv})_k$ is exact in the limit regarding the full dataset

(iv) the subdivision process remains exhaustive.

Proof.

(i) With increasing iteration index k, the dataset $\mathcal{D}_k$ can be augmented or remain constant. The statement follows with any $g^{\text{cv}}(\cdot; \boldsymbol{x}_d, y_d)$ being nonnegative and satisfying the isotonic property.

(ii) The statement follows from Lemma 1.

(iii) Without loss of generality, we focus on a sequence of nested nodes $\{N_{k_j}\}_{j \to \infty}$ generated by the B&B algorithm with growing datasets. Since the augmentation rule completes finitely, there exits $J < \infty : \mathcal{D}_{k_j} = \mathcal{D}\ \forall j \geq J$. Thus, $g_{\mathcal{D}_{k_j}}^{\text{cv}} \equiv g_{\mathcal{D}}^{\text{cv}}$ for any $j \geq J$. The statement follows with $g_{\mathcal{D}}^{\text{cv}}$ being exact in the limit.

(iv) If data augmentation is triggered, a child node is added which coincides with the currently active node except for the dataset. In particular, the parameter domain of the child node equals the parameter domain of its parent node.
If data augmentation is not triggered, the branching process of the original B&B algorithm is executed.
Thus, the subdivision process remains exhaustive.

□

Note that the mathematical formulation of the inequality constraints $h(\boldsymbol{x}_d, y_d; \cdot)$ and $\widetilde{h}$ are not affected by a change of the dataset. We just add more data-dependent constraints $h(\boldsymbol{x}_d, y_d; \cdot)$ when augmenting the dataset. At the latest when reaching the full dataset, we solve the upper bounding problem subject to all inequality constraints of the original problem, see Algorithm 2, and obtain a finite upper bound if the original problem is feasible. Based on Lemma 3, we can extend the proof of Theorem 5.26 [131] to growing datasets.

Theorem 2. *Let Assumptions 4 and 5 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 $h^{cv}(\boldsymbol{x}_d, y_d; \cdot)$ 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 3.8 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.*

Proof. If the algorithm terminates:

- Case I: infinite final upper bound
  Since reducing the dataset results in a relaxation of (PE), the infeasibility of the model follows directly from the proof of Theorem 5.26 given in [131].
- Case II: finite final upper bound
  Since we obtain valid lower and upper bounds even when using a reduced dataset, see Lemma 1, the ε-optimality follows directly from the proof of Theorem 5.26 given in [131].

The finite convergence follows from the proof of Theorem 5.26 given in [131] since

- the modified B&B algorithm generates valid upper and lower bounds for the model based on the full dataset, see Lemma 1,
- Lemma 3 gives exactness in the limit and exhaustiveness of the lower bounding scheme.

□

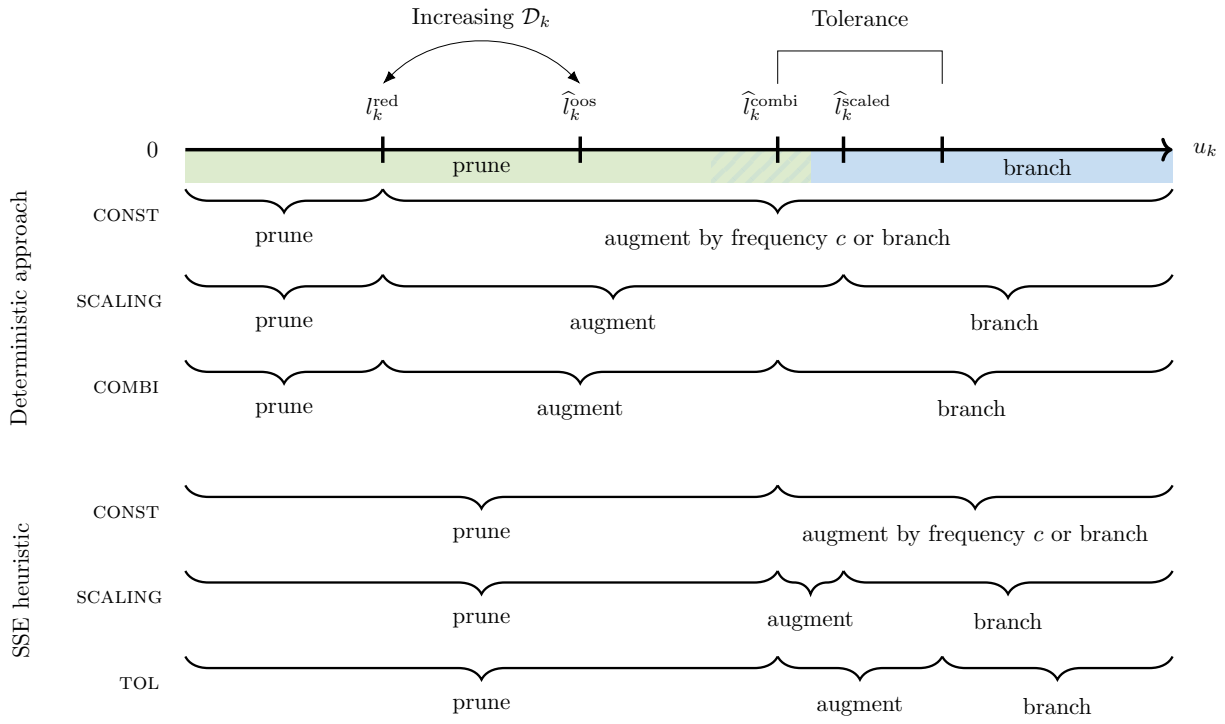


Figure 3.9.: Schematic overview of fathoming a node with objective (SSE) in dependence of its upper bound u_k: pruning, augmenting, and branching based on different augmentation rules as well as the postulated order of reduced, combined, and out-of-sample lower bound. The shaded rectangles indicate the exact fathoming based on the full dataset, where we expect the full lower bound to be in the the hatched region.

Note that the B&B algorithm with growing datasets is only guaranteed to converge if the augmentation rule completes finitely. This is the case, e.g., for augmentation rule $\mathbb{A}_{\text{CONST}}$, where we reach the full dataset with a specific size of the B&B tree, cf. proof of Proposition 1. Otherwise overfitting or conflicting data may prevent augmenting and closing the optimality gap as in the pathological case for augmentation rule $\mathbb{A}_{\text{SCALING}}$ illustrated in Example 1.

In MAiNGO, only nodes with a parameter domain $\mathcal{P}_k$ that has a diameter of at least a user-given tolerance are added to prevent the cluster effect [see, e.g., 54]. In cases of overfitting or conflicting data, MAiNGO may therefore terminate since no active nodes are left, although the optimality tolerance is not reached. Thus, we implemented the following heuristic: we add the node containing the best incumbent found so far with the dataset augmented to the full dataset $\mathcal{D}$. If the solution point of the original problem, i.e., with considering the full dataset, is sufficiently close to the best incumbent calculated based on a reduced dataset, the global solution is contained in the added node. In that case, the B&B algorithm with growing datasets converges successfully. Otherwise the B&B algorithm with growing datasets at least updates the lower bound based on the full dataset and reports therewith tighter bounds on the optimality gap when hitting another termination criterion like a CPU time limit.

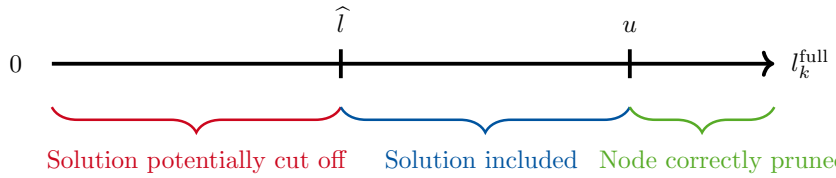


Figure 3.10.: Consequences for the optimal solution in dependence of the relation between post-processed lower bound l_k^{full} and both final lower bound $\widehat{l}$ and final upper bound u reported after termination of the B&B algorithm

3.5. Heuristic pruning

3.5.1. SSE heuristic and post-processing

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 an augmentation rule which does not complete finitely, compare Example 1. Even if the scaled or the combined lower bound is close to the full lower bound, there may be a large range for values of the current upper bound u_k where we augment the dataset although we should just prune the node, see Figure 3.9. In this case, the time savings due to the data reduction may be canceled out by the unnecessary large B&B tree. Motivated by the theoretical and practical results indicating a small distance between the combined and the full lower bound, see Proposition 2 and Postulate 3, we therefore propose the *SSE heuristic* which uses the combined lower bound for *pruning* nodes in the B&B algorithm with growing datasets. Note that SSE refers to using the summed squared prediction error as the objective function.
We can transfer the augmentation rules proposed in Section 3.4.2 from the deterministic approach to the SSE heuristic without any changes, see Figure 3.9. Note that augmentation rule SCALING can hardly trigger augmenting if both the scaled and the combined lower bound are tight approximations of the full lower bounds. Instead, augmentation rule TOL gains importance since it triggers augmenting when we have just missed pruning, i.e., when closing the gap between the full lower bound and its approximation may have a large impact.
Even when the combined lower bound is close to the full lower bound according to Postulate 3, the combined lower bound may be strictly larger than the full lower bound. Hence, we may prune nodes which contain the global solution of the original problem. Indeed, we may make a mistake for any node pruned based on a reduced dataset

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

We therefore propose a post.processing step after the termination of the B&B algorithm, where we calculate the *full* lower bound l_k^{full} of each node $N_k \in \mathcal{N}^{\text{postpro}}$ to detect and quantify any erroneous pruning.
Within 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 3.10. 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 $\min\{l_k^{\text{full}},\ k : N_k \in \mathcal{N}^{\text{changed}}\}$ if $\mathcal{N}^{\text{changed}} \neq \emptyset$ yielding a more accurate value for the true lower bound on the optimal objective value, see Subroutine 2. 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 3.

Subroutine 2 Post-processing

Input: Set of nodes tracked for post-processing $\mathcal{N}^{\text{postpro}}$, final lower bound $\widehat{l}$ after termination of B&B algorithm
Output: Updated final lower bound $\widehat{l}$

for all $k : N_k \in \mathcal{N}^{\text{postpro}}$ **do**
    Solver lower bounding problem (LBP) over $\mathcal{P}_k$ based on full dataset $\mathcal{D}$ and obtain l_k^{full} ;
    if $\widehat{l} > l_k^{\text{full}}$ **then**
        $\widehat{l} := l_k^{\text{full}}$;
    end if
end for

Subroutine 3 Tracking nodes for post-processing within pruning step

Input: Set of nodes tracked for post-processing $\mathcal{N}^{\text{postpro}}$, currently pruned node N with lower bound $\widehat{l}^{\text{combi}}$
Output: Updated set $\mathcal{N}^{\text{postpro}}$

if $|\mathcal{N}^{\text{postpro}}| < 100$ **then**
    $\mathcal{N}^{\text{postpro}} := \mathcal{N}^{\text{postpro}} \cup \{N\}$;
else
    $k_0 := \arg\max_{k:N_k \in \mathcal{N}^{\text{postpro}}} \{\widehat{l}_k^{\text{combi}}\}$;
    if $\widehat{l}^{\text{combi}} < \widehat{l}_{k_0}^{\text{combi}}$ **then**
        $\mathcal{N}^{\text{postpro}} := \mathcal{N}^{\text{postpro}} \setminus \{N_{k_0}\} \cup \{N\}$;
    end if
end if

3.5.2. MSE heuristic

In estimation problems from the 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) as it is often more stable regarding the size of the dataset. 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 [2]. To test whether these changes improve the performance of the B&B algorithm with growing datasets, we propose the *MSE heuristic* on top of the SSE heuristic by replacing objective (SSE) with (MSE) in (PE).
Let $\boldsymbol{p}_{k,\mathrm{MSE}}^{\mathrm{lb,full}}$ and $\boldsymbol{p}_{k,\mathrm{MSE}}^{\mathrm{lb,red}}$ be the optimal solution point 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 lower and reduced lower bound with $l_k^{\mathrm{full,MSE}}$ and $\widehat{l}_k^{\mathrm{red,MSE}}$. Note that the reduced lower bound may be larger than the full lower bound or even the optimal objective value when using (MSE) and is therefore denoted as a heuristic lower bound.
Additionally, we adapt the definition of the heuristic out-of-sample lower bound yielding

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

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.

Proposition 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 Proposition 2, we obtain

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

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

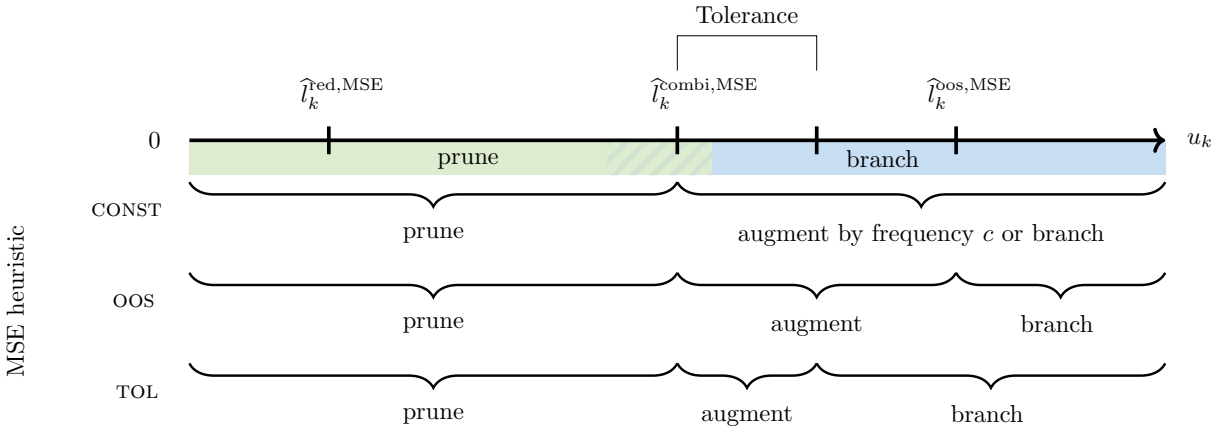


Figure 3.11.: Schematic overview of fathoming a node with objective (MSE) in dependence of its upper bound u_k: pruning, augmenting, and branching based on different augmentation rules as well as the postulated order of reduced, combined, and out-of-sample lower bound. The shaded rectangles indicate the exact fathoming based on the full dataset, where we expect the full lower bound to be in the the hatched region.

Proposition 3 shows that we can expect that the combined lower bound

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

is 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 3, we expect the following.

Postulate 4.

(i) For most B&B nodes, the difference between combined lower bound $\widehat{l}_k^{combi,MSE}$ and full lower bound $l_k^{full,MSE}$ is vanishingly small.

(ii) 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 2 and 3 with the adapted definitions of the combined and full lower bound.
We re-use augmentation rules CONST and TOL for the MSE heuristic, compare Figure 3.11. Since augmentation rule 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

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

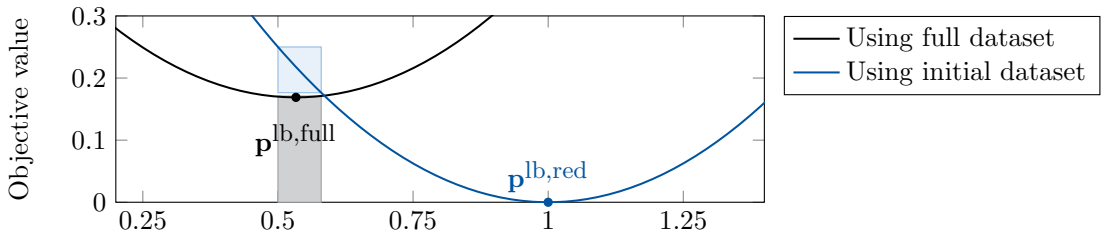


Figure 3.12.: Objective function and global solution of the parameter estimation problem described in Example 3 based on different datasets with the domain containing the optimal solution calculated via constraint propagation based on the full dataset ( ) and the corresponding optimum domain based on the initial dataset ( )

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.
Note that we should not apply augmentation rule OOS for the deterministic approach and the SSE heuristic, since we expect the out-of-sample lower bound (i) to be smaller than the full lower bound in most of the nodes and (ii) to be even smaller than the reduced lower bound when closing up to the full dataset, see Postulate 2. In case (i), we may blow up the B&B tree unnecessarily by branching nodes which should be pruned or for which we should at least trigger augmentation. In case (ii), we can never reach the full dataset since augmenting could only be triggered for nodes which have been pruned already.

3.5.3. Impact on bound tightening heuristics

Bound tightening heuristics like constraint propagation or optimality-based bound tightening [79] can accelerate the B&B algorithm by narrowing down the domain containing the global solution (*optimum domain*). However, bound tightening heuristics may fail when comparing optimum domains calculated based on different datasets as shown in the following example.

Example 3 (Bound tightening fails for reduced mean squared error). We revisit the estimation problem of Example 1 using the mean squared error as the objective function

$$\min_{p\in[0,25]} \frac{1}{|\mathcal{D}|} \sum_{(x_d,y_d)\in\mathcal{D}} (p \cdot x_d - y_d)^2$$

with full dataset $\mathcal{D} = \{(1,0),(1,0.6),(1,1)\}$ and the global optimal value 0.1689 at solution point $p^* = 0.5333$.
Assume that constraint propagation in the root node based on the full dataset can narrow down the feasible domain to interval $p \in [0.50, 0.58]$. At the beginning of the B&B algorithm, we have the natural lower bound of 0 and the evaluation at $p = 0.58$ as the best upper bound found so far. Thus, the optimal solution is guaranteed to be within the domain $(p,u) \in [0.50, 0.58] \times [0, 0.1711]$, see lower box ( ) in Figure 3.12.
Before entering the B&B algorithm, the dataset in the root node is replaced by the initial dataset, see Algorithm 2. Assume that the initial dataset is given by $\mathcal{D}_0 = \{(1,1)\}$. Correspondingly, bound tightening based on the initial dataset via interval arithmetic

suggests that the global solution is contained in the domain $[0.50, 0.58] \times [0.1764, 0.25]$, see upper box ( ).
Since the two domain boxes do not intersect, we may erroneously conclude that the root node cannot contain the global solution and, therefore, that the problem is infeasible. □

In general, we need the feasible objective values of a node to lie within the lower and upper bound of its parent node even when changing the dataset to guarantee the correctness of bound tightening techniques. The dataset is (*i*) reduced between pre-processing the root node and entering the B&B algorithm, cf. Algorithm 2, and (*ii*) increased when augmenting. In case (*i*), we need the feasible objective values given by the initial dataset to lie within the natural lower bound of 0 and an upper bound found based on the full dataset. While the summed squared prediction error can only decrease or stay the same when reducing the dataset, the mean squared error may increase as shown in Example 3. In case (*ii*), we need the lower and upper bound of a node to contain the feasible objective values calculated based on an augmented dataset, i.e., we need valid lower and upper bounds for larger datasets. In our implementation of the B&B with growing datasets in MAiNGO, we consequentially allow bound tightening heuristics only for the deterministic approach and the SSE heuristic in pre-processing and only for the deterministic approach in nodes considering a reduced dataset.

3.6. Numerical performance of the B&B algorithm with growing datasets

3.6.1. Computational performance for single B&B nodes

Before studying the performance of the B&B algorithm with growing datasets for the DGO of parameter estimation problems, we investigate the time gained in the B&B nodes due to data reduction. With this, we extend the study presented Section 3.3 from subregions of the parameter domain representing artificial nodes to actual B&B nodes obtained within the B&B algorithm. Again, we focus on the EOS model discussed in Section D.3.2.1, but, in this section, based on two full datasets with $|\mathcal{D}| = 262$ and $|\mathcal{D}| = 2262$ yielding problems EOS262 and EOS2262.
All calculations for the study in this section are performed on an Intel(R) Core(TM) i5-3570 processor with 3.4 GHz and MAiNGO version 0.7.2. For the lower bounding, we apply an adaptation of Kelley's algorithm [109, 147] for linearization and invoke the linear solver CLP version 1.17.0 [69] to solve the resulting linear program. For the upper bounding, the local solver LBFGS [129, 148] implemented in the NLOPT toolbox v2.5.0 [103] is used. We use the deterministic approach of the B&B algorithm with growing datasets.
In each node of the standard B&B algorithm, we solve the lower and upper bounding problem as well as bound tightening techniques as a pre- and post-processing for the bounding procedures. While the pre-processing comprises bound tightening via constraint propagation and optimality-based bound tightening [79], the post-processing contains of duality-based bound tightening [166]. When using the B&B algorithm with growing datasets, we further need to update the optimization problems for the lower and upper bounding to apply the (reduced) dataset associated to the processed node. As depicted in Figure 3.13, re-building the lower and upper bounding problem may take as long as solving the lower

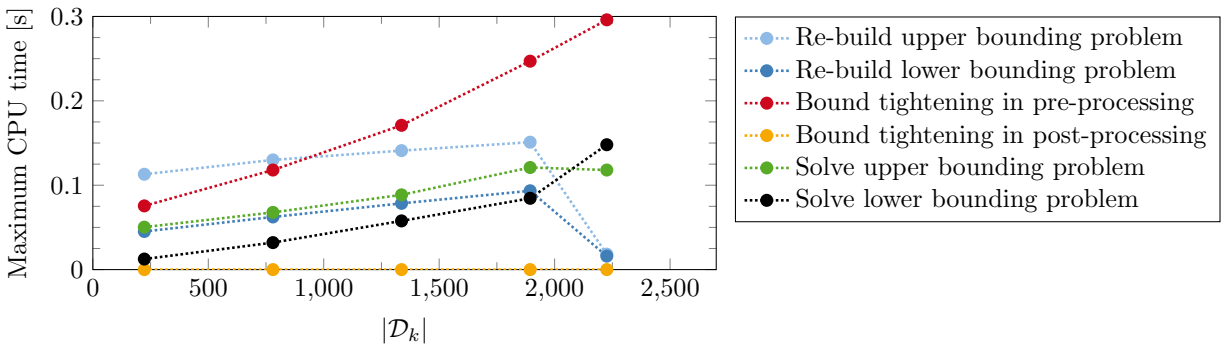
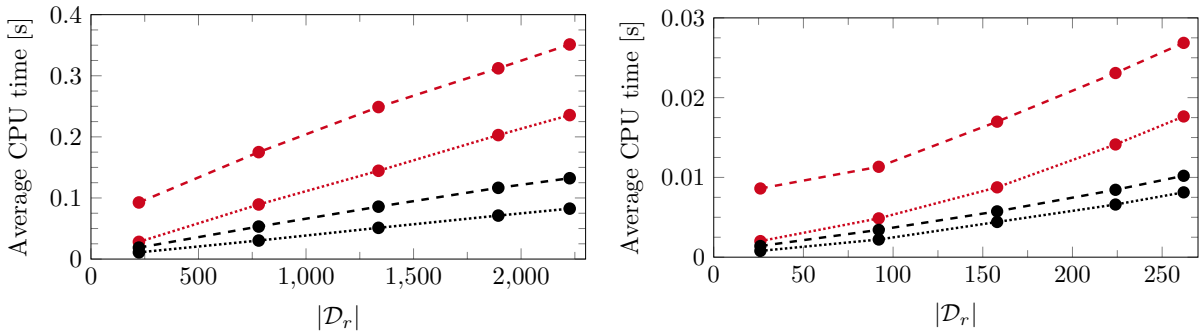


Figure 3.13.: Maximum CPU time spent by the deterministic approach of the B&B algorithm with growing datasets implemented in MAiNGO for the different tasks when processing a node within the DGO of problem EOS2262, where the B&B nodes are clustered by the size of their associated dataset

or upper bounding problem in the worst case. That's why we consider only a few different datasets between which we may switch frequently. In detail, we re-build the bounding problems only once for each (reduced) dataset and save a copy which can be efficiently accessed by any other node considering the same dataset. With this, the average computational costs for updating the lower and upper bounding problem are negligible, compare the last time measurements for $|\mathcal{D}| = 2262$ in Figure 3.13. These measurements show the computational effort for re-using a copy of the respective bounding problems based on the full dataset. Similarly, the computational effort for post-processing a node is vanishingly small. We observe hat the maximum CPU time for both pre-processing and solving the lower bounding problem increases about linearly with increasing size of the dataset, which is consistent to the preliminary findings in Section 3.3, compare Figure 3.13 with Figure 3.5.

As hinted when introducing the standard B&B algorithm in Section 3.2.2, MAiNGO uses McCormick relaxations to calculate convex underestimators for the lower bounding problem, whereas widely used DGO solvers like ANTIGONE, BARON, and SCIP apply the auxiliary variables method (AVM). While the number of optimization variables treated within the B&B algorithm remains constant by McCormick relaxations, AVM adds more auxiliary optimization variables if more (nonlinear) terms are included in the model functions and, thus, if a greater number of data points is considered. Aside from that, the use of McCormick relaxations makes MAiNGO particularly suitable for reduced-space formulations [32, 140], meaning again that the number of constraints and optimization variables does not necessarily grow with the size of the dataset, cf. also the introductory paragraph of Appendix D.3. In other words, the size of the dataset mainly affects the computational costs for function evaluations rather than the size of the optimization problem solved. Consequently, we expect that the extension to growing datasets is even more advantageous for AVM-based solvers like BARON, especially when working with full-space formulations within these. However, a general study on the performance of different DGO solvers as well as implementing the proposed extension in third-party DGO solvers is not in scope of this thesis since these solvers are either not open-source or have a too complex code structure.

We use the feature for adding auxiliary variables in MAiNGO [147] to get nevertheless a first



(a) EOS2262 **(b)** EOS262

Figure 3.14.: Average CPU time spent by the deterministic approach of the B&B algorithm with growing datasets implemented in MAiNGO for pre-processing (•) and solving the lower bounding problem (•) without (dotted lines) and with auxiliary variables (dashed lines) in a node, where the B&B nodes are clustered by the size of their associated dataset

impression on potential CPU time savings when extending an AVM-based B&B algorithm by growing datasets: for each mathematical expression that occurs at least two times in the model formulation, an auxiliary optimization variable is added to the lower bounding problem. In case of EOS2262 49 auxiliary variables are added, while 17 auxiliary variables are added for EOS262. Since the upper bounding problem is not affected by these auxiliary variables and the CPU time for changing the lower and upper bounding problem as well as for post-processing is comparatively small, we focus solely on the CPU time for pre-processing and solving the lower bounding problem in the following paragraph.
Even without auxiliary variables, an increasing size of the dataset implies an increasing effort for evaluating data-dependent functions and their subgradients, see Figure 3.14. The addition of auxiliary variables shifts the CPU times to larger values. In MAiNGO, the number of optimization variables within the B&B algorithm remains the same for each dataset even when using the auxiliary variable feature. Thus, we expect both a shift and a larger slope of the curves when adding one or more auxiliary variables per data point. Note that the seemingly small CPU time differences add up to significant time savings since the B&B algorithm often requires the processing of multiple hundred thousands of nodes for convergence, cf. Figure 3.16.

3.6.2. Benchmarking 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 Section 3.4 as well as the heuristic approaches from Section 3.5 using different augmentation rules. We run each of the approaches with and without a resampling heuristic, see detailed results in Section 3.6.3.
Many real-world parameter estimation problems minimize a summed squared error over 1 000, 10 000 or more measurements [e.g., 8, 107, 124]. Contrarily, general benchmark libraries for mathematical programming like MINLPlib [37] and PrincetonLib [202] offer only limited coverage of optimization problems in this format or only small problems. For example, the COCONUT benchmark [185] contains a few instances with up to 35

Table 3.5.: Overview on general properties and references of the models and data used in the case study

Name of estimation problem	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	[124, 174]	[175]
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	[8]	[56, 57]
IHMunc	10	284	1	measured	NLP	— ” —	— ” —
kinetics	5	446	1	measured	DO[a]	[140, 187, 197]	[140]
EIS	7	26	1[b]	measured	NLP	[171]	[198, 199]
Biochemistry							
TSPexact	12	20	8	synthetic, exact	DO[a]	[142, 206]	[175]
TSPnoisy	12	20	8	synthetic, noisy	DO[a]	— ” —	— ” —
Machine learning							
GMMcon	6	272	1	measured	NLP	[49]	[13, 88]
GMMineq	5	272	1	measured	NLP	— ” —	— ” —
trainANN	21	220	3	measured	NLP	[95, 181]	[62, 63]

[a]Transformed into an NLP by applying the explicit Euler scheme to the differential equations

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

data points. The largest instance named “gulf” [44, 143] containing 99 data points can be solved within a second even without growing datasets. At the same time, the second largest instance with 65 data points, see “osborneb” or “Osborne 2” [143, 150], containing 11 optimization variables and redundant terms in the objective poses a huge challenge for the local upper bounding solver which is not tackled by the algorithm proposed in this thesis. Benchmarks from medicine and biology [e.g., 206] typically include large datasets but pose a huge challenge for DGO solvers as they contain large dynamic optimization problems with about 30, 100, or more unknown parameters.

To obtain an informative benchmark study, we therefore collected different models from both the 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 parameter and a different number and dimension of data points in full dataset $\mathcal{D}$, see Table 3.5. The exact mathematical expressions are provided in Appendix D.3. For ready-to-use implementations of the model as well as the exact data points used, refer to the openly accessible repository GloPSE, cf. Section 3.1.3.

All computations are performed on Intel Xeon Platinum 8160 processors “SkyLake” (frequency 2.1 GHz, RAM = 3.75 GB) with our open-source solver MAiNGO v0.8.0. Even small deviations in computational times for processing one node may significantly change the number of nodes processed within our CPU time limit of 23 h 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 23 h such that we can guarantee a total runtime of 24 h including the initialization of the parameter estimation problem 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 problem specific values $\varepsilon^A = 0.01/|\mathcal{D}|$ when using (MSE). In each optimization run, we use Ipopt version 3.12.12 [210] for running 3 local searches as a pre-processing step in the root node. Moreover, we use McCormick relaxations and a linearization in the midpoint of the parameter intervals to obtain a linear program (LP) for lower bounding. For estimation problems IHMcon, IHMunc, EIS, TSPexact, TSPnoisy, GMMcon, GMMineq, and trainANN, we solve the LP with linear optimizer CLP v1.17.0 [68]. We use interval extensions [118] calculated with FILIB++ [125] instead for problems EOS262, EOS2262, EOS262noisy, EOS2262noisy, as the resulting LP of models EOS and kinetics are apparently numerically difficult for CLP and alternative LP solver CPLEX [101], compare also THE first paragraph of Section 3.3.2. When using pure interval extensions, we disable the optimality-based bound tightening. The upper bounding problem is solved with local optimizer LBFGS [129, 148] implemented in the NLOPT toolbox v2.5.0 [103] for all estimation problems. Since the quality of the local solutions is very sensitive to the start solution chosen and the number of steps performed in the local optimization of the upper bounding problem, we limit the local optimization procedure in both pre-processing and upper bounding by a maximum number of iterations rather than a CPU time limit to get deterministic and comparable results. The complete settings used are provided in Appendix D.1.

3.6.2.1. Post-processing statistics

At first, we study the post-processing step to check whether we can expect the heuristic approaches to converge to the global solutions. Table 3.6 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 problems IHMcon, IHMunc, EIS, and trainANN are not listed, just as the statistics of the MSE heuristic for the TSP model, 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. Note that we also achieve $|\mathcal{N}^{\text{postpro}}| < 100$ in the rare cases where we need to prune less than 100 nodes for convergence, namely when solving problem TSPexact with the SSE heuristic applying augmentation rule TOL.
We observe further that for only 4 out of 13 problems there are nodes where the lower bound l_k^{full} calculated in post-processing falls below the final upper bound, i.e., which are pruned although they may contain the global solution. Out of these, the final lower bound is updated solely when using the SSE heuristic applying augmentation rules SCALING

Table 3.6.: Post-processing statistics for the problems 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 $\widehat{l}$ was changed as well, and the maximum CPU time over all augmentation rules

Problem	Res.?[a]	$\lvert\mathcal{N}^{\text{postpro}}\rvert$	No. of nodes wrongly pruned[b]	$\widehat{l}$ changed?[c]	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
TSPexact	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
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] Res. = Resampling heuristic used, see Section 3.6.3

[b] Nodes with $l_k^{\text{full}} \leq u$, see Section 3.5.1

[c] 0 = no, 1 = yes

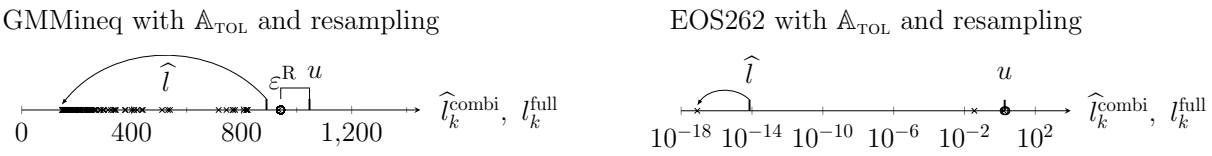


Figure 3.15.: Lower bounds of nodes in $\mathcal{N}^{\text{changed}}$ before ($\circ$) and after ($\times$) post-processing and change of final lower bound $\widehat{l}$ due to post-processing compared to final upper bound u and relative optimality tolerance ε^{R}

Table 3.7.: Total CPU time needed for convergence of the B&B algorithm. Only problems 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 problem.

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

[a]Hitting CPU time limit of 23 h.

and TOL for problem EOS262 and augmentation rule TOL for problems GMMcon and GMMineq. We pick two exemplary cases to depict the changes due to post-processing in Figure 3.15. 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 problem EOS262, the changes due to post-processing are essentially nonexistent: we find numerically insignificant differences between 10^{-14} and 10^{-18}. In contrast to that, the final lower bound of problems GMMcon and GMMineq change significantly, e.g., 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 problems 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 proposed post-processing procedure is very cheap: over all 5 repetitions of the case study we measured a maximum runtime of 11.3 s.

3.6.2.2. Computational performance

We proceed with studying the general performance of the B&B algorithm with growing datasets compared to the standard B&B algorithm as a benchmark. Only 4 problems

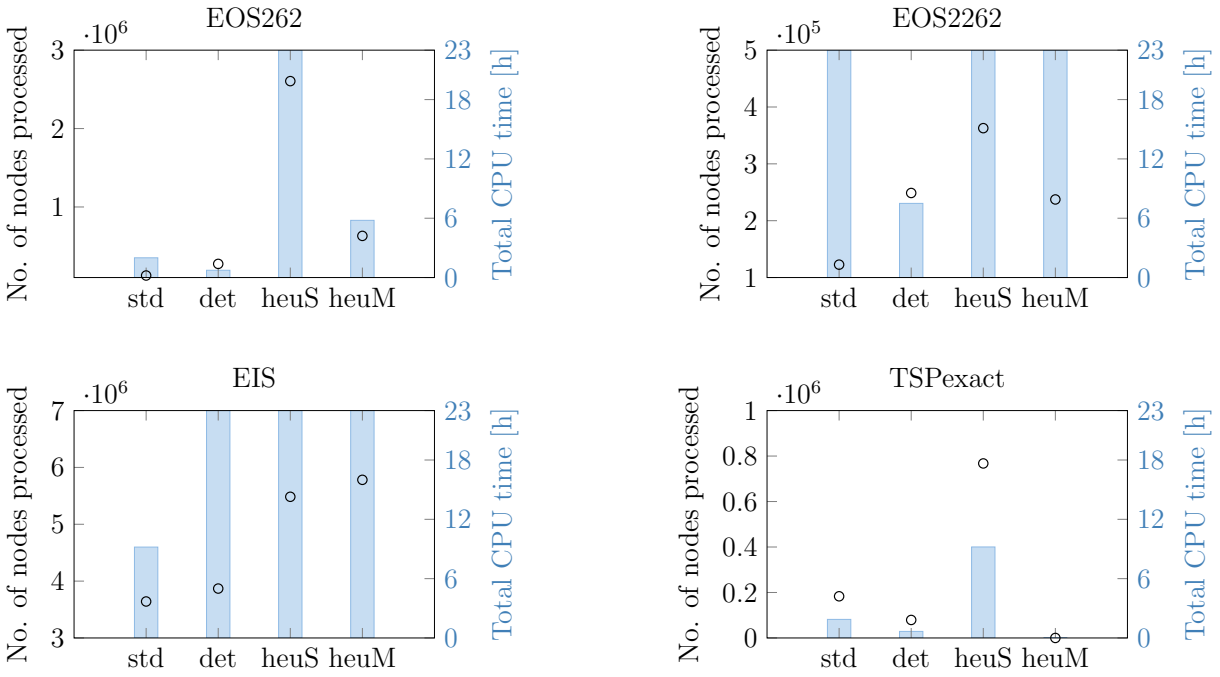


Figure 3.16.: Number of nodes processed (o) and total CPU time used (■) by the standard B&B algorithm with (SSE) as well as the deterministic approach applying augmentation rule SCALING (det), the SSE heuristic (heuS), and the MSE heuristic (heuM), both applying augmentation rule TOL, of the B&B algorithm with growing datasets without the resampling heuristic for different problems when running DGO with a CPU time limit of 23 h

converge within the CPU time limit of 23 h for at least one of the approaches including all 3 problems with exact data, namely EOS262, EOS2262, and TSPexact, compare Table 3.7. For problems 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. In general, the time needed for convergence depends on (*i*) the processing time for a node and (*ii*) the number of nodes needed for convergence. The strength of the B&B algorithm with growing datasets lies in the significant reduction of the CPU time per node, cf. Section 3.6.1. Figure 3.16 shows that we can decrease the CPU time required for convergence of problems EOS262 and EOS2262 compared to the standard B&B algorithm when using the deterministic approach while processing more nodes in total. In these cases, the major part of the B&B tree is fathomed based on reduced datasets. For (*ii*), the tightness of the lower and upper bounds plays a key role. For exact data, the final lower bound equals the natural lower bound of 0 except for numerical tolerances. The upper bound depends on the actual local solution found. We calculate the solution point of the upper bounding problem based on the reduced dataset, see Algorithm 2, while, in MAiNGO, the solution point of the lower bounding problem is used for initializing the upper bounding solver. By chance, a reduced dataset may therefore give a better solution and lead to much faster convergence as can be seen for problem TSPexact when comparing the deterministic approach with the standard B&B algorithm. When using the MSE heuristic to optimize problem TSPexact, the global solution is even found in the root node, resulting in a runtime of 2 s, while the standard B&B algorithm hits the CPU time limit with objective (MSE).

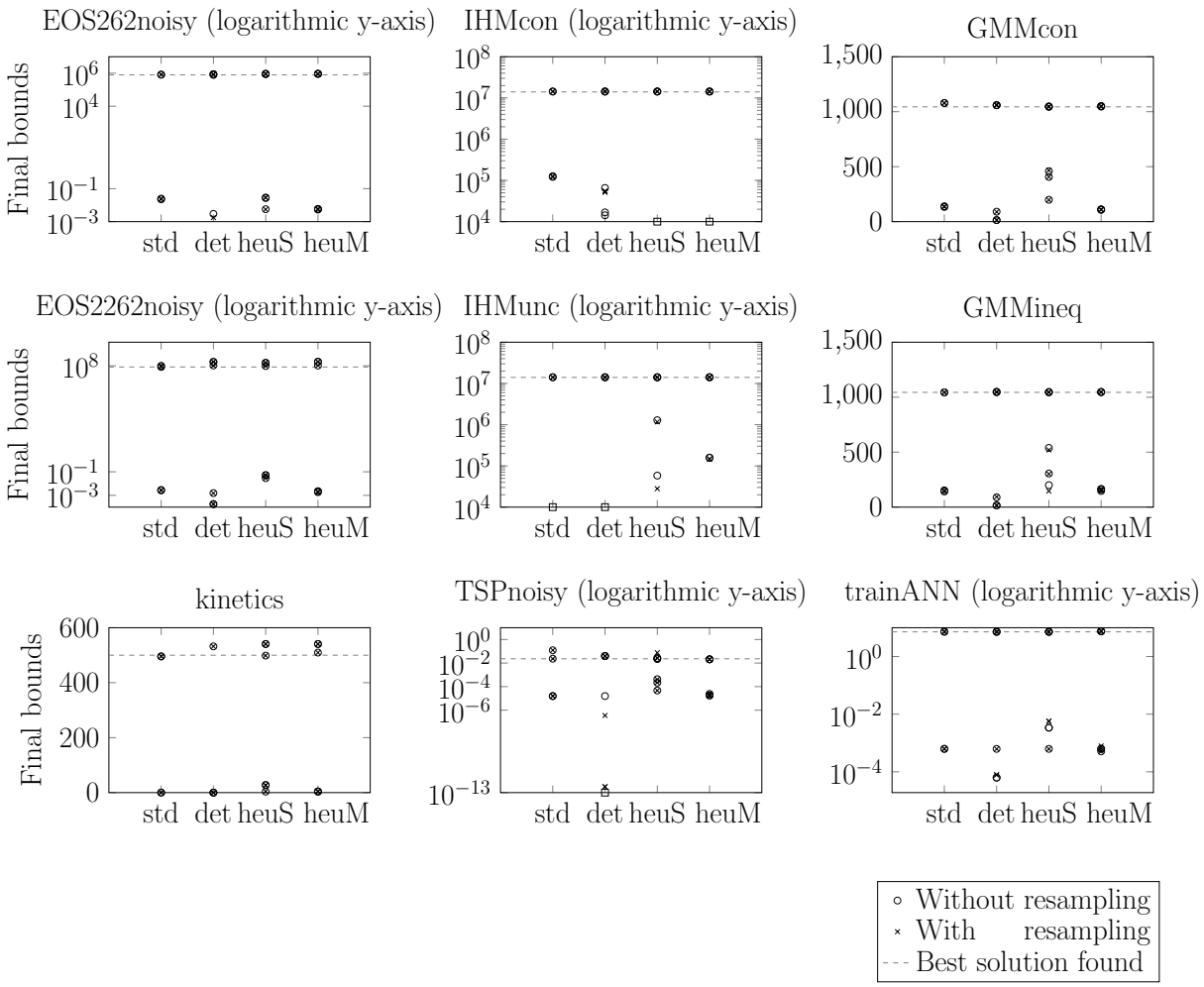


Figure 3.17.: Final lower bounds (strictly below dashed line) and final upper bounds (above dashed line) on the optimal objective values of the problems which did not converge within a CPU time limit of 23 h for any of the approaches, namely the standard B&B algorithm (std) with objectives (SSE) and (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 3.9 and 3.11. If applicable, the final lower bounds are subject to post-processing. The squares in the logarithmic plots indicate lower bounds equal to 0.

When using noisy data, the tightness of the lower bound may suffer from the data reduction, which is the downside of the deterministic approach. For example, the standard B&B algorithm converges within 8 h for problem EIS, whereas all approaches of the B&B algorithm with growing datasets hit the CPU time limit. More specifically, the standard B&B algorithm processes within 8 h about the same amount of nodes than the deterministic approach, although all nodes processed by the deterministic approach are associated with the initial dataset, see Figure 3.16. Note that problem EIS considers with $|\mathcal{D}| = 26$ as well as $\dim(\boldsymbol{x}) = 1$ and $\dim(y) = 2$ a small three-dimensional dataset for fitting 7 unknown parameters. By default, we consider an initial dataset with $|\mathcal{D}_0| = 3$. Based on this initial dataset, we cannot prune any node. As a consequence, the number of active nodes $\mathcal{N}_k$ increases monotonously with each B&B iteration k which increases the effort for handling the B&B tree, i.e., for updating the (ordered) sets $\mathcal{N}_k$ and $\mathcal{N}_k^{\mathrm{f}}$ as well as for updating the lower bound over all active nodes. Note that neither the overall lower nor the best upper bound changes for the case depicted in Figure 3.17, namely when solving problem EIS with the deterministic approach applying augmentation rule SCALING.
For all problems which do not converge within 23 h, all approaches give a similar or even the same optimal objective value after the CPU time limit, compare the values above the dashed lines in Figure 3.17. As already seen when discussing the post-processing, the SSE heuristic tends to give significantly larger, i.e., better, lower bounds. While the SSE heuristic and the standard B&B algorithm perform similarly only for EOS262noisy, the standard B&B algorithm finds the best lower bound only for IHMcon. Note that the lower bounds presented in Figure 3.17 include the correction performed in the post-processing step, see Appendix D.2 for the exact values.
We recall that the models of problems 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 [32, 140]. 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 problem IHMunc is in order of magnitude 10^6, while the best lower bound for problem 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 study, the reduced-space formulation GMMineq is slightly advantageous for finding a good upper bound, while the lower bounds obtained for problems GMMcon and GMMineq are similar for the different approaches.

3.6.2.3. Comparison of the augmentation rules

Before comparing the performance of the different augmentation rules, we recall the following concerning their theoretical background: Firstly, we have proven finite convergence when using augmentation rule CONST, see Proposition 1 and Theorem 2, but cannot make statements about the respective speed of convergence. Secondly, we have a high chance to augment where needed when using augmentation rule SCALING, see Theorem 1. Thirdly, augmentation rules SCALING and OOS share the same motivation but use different approximations of the full lower bound. Finally, augmentation rule TOL is designed to trigger

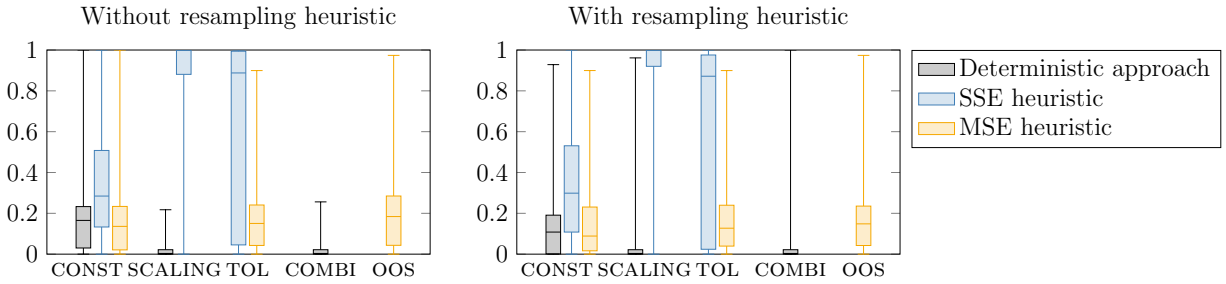


Figure 3.18.: Box-Whisker-plot of the final lower bounds normalized with the maximum final lower bound achieved for the respective problem in dependence of different augmentation rules for the different approaches of the B&B algorithm with growing datasets. If applicable, the final lower bounds are updated within post-processing.

augmentation for nodes where we had just missed pruning. In this section, we focus on the final lower bounds since the final upper bounds are almost invariant to the specific approach used, compare Figure 3.17. As the order of magnitude of the final lower bounds varies largely for the different problems, we normalize the values by division with the maximum final lower bound achieved for the respective problem.
Figure 3.18 illustrates how the choice of the augmentation rule affects the normalized final lower bounds. When combining SCALING or COMBI with the deterministic approach of the B&B algorithm with growing datasets, we achieve the smallest, i.e., worst, lower bounds in average. Table 3.7 shows that this is no general rule: When solving problems EOS262, EOS2262, and TSPexact with the deterministic approach, augmentation rule SCALING yields the fastest convergence behavior. Similarly, augmentation rule SCALING performs best for the SSE heuristic. When using augmentation rule TOL instead, we may obtain just as good or significantly worse results. Augmentation rule CONST gives the smallest lower bounds in average when using the SSE heuristic. For the MSE heuristic, all three tested augmentation rules perform similarly.
In summary, the right choice of the augmentation rule depends on the specific parameter estimation problem. In this case study, we obtain overall the tightest final lower bounds when combining augmentation rule SCALING with the SSE heuristic. Note that augmentation rules CONST, SCALING, and TOL may be further tuned by changing frequency c, augmentation weight ρ, and augmentation tolerance $\widehat{\varepsilon}$, respectively.

3.6.3. Insights regarding 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 [2, 144]. In the context of this thesis, we may interpret (i) the DGO based on the full dataset $\mathcal{D}$ as training and any unseen or even unknown data points exceeding $\mathcal{D}$ as new data or (ii) solving the reduced problem based on reduced dataset $\mathcal{D}_k$ as training and the data points contained in complementary set $\mathcal{D} \setminus \mathcal{D}_k$ as new data. Regarding case (i), we note that the full dataset is treated as a fixed part of the parameter estimation problem in this thesis, cf. Section 3.2.1. In particular, the global optimal solution may provide a closer fit to the given noisy dataset than to exact data obtained by evaluating the model equations, see, e.g., the optimal solutions for model TSP discussed in Appendix D.3.7.2. This does not

necessarily lead to overfitting. For example, the solutions of model trainANN generalize well to data not included in the full dataset, cf. Appendix D.3.6.2. A further study on uncertainty quantification regarding noisy data is out of scope of this thesis since we assume a non-random full dataset, see Assumption 6.
On the one hand, we tackle case (*ii*) by using out-of-sample evaluations for obtaining a better approximation of the full lower bound, cf. Section 3.4.2.3. On the other hand, we can apply resampling to improve the reduced lower bound used for both the heuristic augmentation rules and the heuristic pruning. In machine learning and statistics, resampling techniques like cross-validation, random forests, boosting, and bootstrap aggregation are used to reduce the estimation variance, see [58, 159], [102, Chapter 5 and Section 8.2] and [93, 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 [35]. We transfer this property to the aggregation of reduced lower bound using bootstrap aggregation or, in short, *bagging* [34]. 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 $g^{\text{cv}}_{\mathcal{D}_{k,*}}(\boldsymbol{p}_k^{\text{lb,red},*})$, and aggregate the resulting reduced lower bounds. Since the computational effort for solving the lower bounding problem is approximately linear in the size of the used dataset, see Section 3.6.1, using two subsamples already means doubling the computational effort. We therefore restrict ourselves to a maximum of two subsamples. It can be shown that the expected value of the reduced lower bound remains the same when applying bagging, whereas we can half the variance when using bagging with two independent subsamples. However, the larger the reduced datasets $\mathcal{D}_{k,1}, \mathcal{D}_{k,2} \subsetneq \mathcal{D}$ representing the subsamples, the higher the chance of shared data points $|\mathcal{D}_{k,1} \cap \mathcal{D}_{k,2}| > 0$ and, thus, of dependency. We therefore propose to resample only the initial dataset. Altogether, we obtain a *resampling heuristic* which replaces the reduced lower bound l_k^{red} of any node considering the initial dataset $\mathcal{D}_k = \mathcal{D}_0$ by the arithmetic mean

$$\frac{l_k^{\text{red}} + l_k^{\text{red},*}}{2} ,$$

where lower bound $l_k^{\text{red},*}$ is calculated based on another random subset $\mathcal{D}_{0,*} \subsetneq \mathcal{D}$ with size $|\mathcal{D}_{0,*}| = |\mathcal{D}_0|$.
When comparing the CPU times obtained in the benchmark case study with and without resampling, cf. Table 3.7, 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 3.17. Consequentially, the performance of the augmentation rules does not change significantly when using the resampling heuristic, see Figure 3.18. Also, 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 problem EOS2262. 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. [34] and [102, Section 8.2].

3.7. Conclusions and outlook

In this chapter, we proposed the B&B algorithm with growing datasets to make the solution of parameter estimation problems based on large datasets (better) tractable for deterministic global optimization. In detail, we start with a subset of the dataset in the root node and let it grow, i.e., augment data points, until converging to the full dataset provided with the model formulation. By reducing the estimation problem in this way, we can save computational effort for processing the nodes of the B&B tree. For deciding when to augment the dataset, we propose different *augmentation rules*. We have proven that the B&B algorithm with growing datasets converges towards the global optimum of the original estimation problem regarding the full dataset.
The lower bound calculated based on a reduced dataset (*reduced lower bound*) may be significantly smaller than the lower bound based on the full dataset (*full lower bound*). In particular for error-prone measurements, this gap between reduced and full lower bound may prevent us from pruning nodes and, thus, compensate any time savings obtained by solving a reduced problem. Hence, we proposed the SSE and MSE heuristic which use an approximation of the full lower bound for pruning nodes. For this approximation, we adapt the reduced lower bound with the help of out-of-sample evaluations to decrease the effects of overfitting. As the approximated lower bound may be larger than the full lower bound or even larger than the global optimum, we obtain a heuristic pruning procedure. To detect and quantify any potential mistakes due to the heuristic pruning, we propose a post-processing procedure to update the final lower bound after termination of the B&B algorithm.
Note that we investigated parameter estimation problems minimizing the summed squared error or the mean squared error of the model predictions. But the presented proof of convergence for the B&B algorithm with growing datasets holds for any objective summing up non-negative continuous measures of the model-data mismatch for the single data points. This allows for the inclusion of non-negative weights, compare, e.g., the scaling of the predictions errors with the respective measurements in models EIS and EOS as presented in Appendix D.3.1.1 and D.3.2.1. Moreover, the proof can be applied as well when adding a penalization term based on the estimated parameters to the objective, e.g., for regularization, compare Appendix D.3.6.1, or for favoring sparse solutions [e.g., 193]. Instead of minimizing the least squares deviation (l_2-estimation), we may minimize the sum of negative log-Likelihood functions (maximum likelihood estimation), compare model GMM presented in Appendix D.3.3.1, the sum of the absolute prediction errors (l_1-estimation), or the sum of other l_p-norms as suitable for the specific measurement dataset considered [e.g., 161]. In particular, we may transfer measures for prediction errors which are more robust to outliers from statistics, see, e.g., [99, 100] and [211, Chapter 16]. Note that when using measures which may decrease when adding data points, like the median of squares [164] proposed for robust regression, the lower bounds calculated based on the reduced dataset may not be valid for the original problem, just as when minimizing the mean squared error.
We investigated the numerical performance of the different approaches of the B&B algorithm with growing datasets with the standard B&B algorithm as a benchmark based on 13 real-world parameter estimation problems from both the literature and colleagues. For the estimation problems with exact measurement data, the deterministic approach of the B&B algorithm can obtain faster convergence compared to the standard B&B algorithm. For the problems with noisy measurement data, the SSE heuristic yields overall the largest,

i.e., best, final lower bounds within the given CPU time limit. The best objective value found is almost invariant to the choice of the algorithm. For some of the problems, we find the best solution even before entering the B&B algorithm when running a multi-start of local searches in the root node as a pre-processing. Thus, we expect stochastic approaches and metaheuristics for global optimization to give similar results. However, only after convergence of the DGO method we get a theoretical guarantee for the optimality of the solution. For the problems where we wait for the upper bound to converge, integrating an implementation of the generalized Gauss-Newton algorithm [76, 137, 178] is of high interest since it has favorable convergence properties for the local optimization of parameter estimation problems [28].

Overall, we can significantly enhance the computational performance of the spatial B&B algorithm when using growing datasets. Since we use reduced-space formulations for the models as well as McCormick relaxations for computing the lower bounding problem, the number of optimization variables is invariant to the size of the dataset. Due to the reduced computational effort for function evaluations, evaluations of subgradients, and, thus, computing the linearizations themselves, we observe linear time savings for processing a node in dependence of the size of the dataset. We have shown that the average CPU time for processing a node rises significantly when adding a fixed number of auxiliary optimization variables. We conclude that the B&B algorithm with growing datasets is even more advantageous when using a data-dependent number of auxiliary variables. Hence, including the B&B algorithm with growing datasets in DGO solvers like ANTIGONE, BARON, and SCIP which apply the auxiliary variables method has a great potential for future research.

We aim also at extending the applicability of the B&B algorithm with growing datasets to models with data-dependent constraints. On the one hand, this allows to replace data-dependent model equations with equality constraints on auxiliary variables for handling, e.g., matrix inversions for the optimization of Gaussian processes [182] or full-space formulations, see, for example, the differences between models IHMcon and IHMunc as well as GMMcon and GMMineq. On the other hand, this allows to pose physical constraints on each of the measurements as required, e.g., for fitting binary fluid systems [30] and for simultaneously optimizing the baseline and Pseudo-Voigt functions of the IHM model discussed in Section D.3.4. While data-dependent constraints are already considered in the theoretical findings of this thesis, a specific treatment of such constraints has not been implemented yet.

The number of data points used for the reduced datasets can be easily adjusted via a user setting. If the full dataset is small compared to the number of unknown variables, the default value for the size of the initial dataset should be adjusted. As a rule of thumb, the *1 in 10 rule* from statistics may be used which states that the number of data points should be (at least) about 10 times the number of unknown parameters [90, 91, 154, 194].

It remains for future studies to enhance the choice of the specific data points used for both the initial dataset and augmentation. As is, we randomly draw indices from range 1 to $|\mathcal{D}|$. Instead, we may also use Latin hypercube sampling in the space of the data points. As an alternative, we may provide a (somehow) sorted dataset and choose the data points sequentially, compare [122]. On the one hand, this moves the computational effort for the random drawing out of the B&B algorithm into a data preparation phase when using a shuffled dataset. On the other hand, this allows to exploit a-priori knowledge on the data, e.g., by

- designing a specific initial dataset which is known to be sufficient for representing the general properties of the physical model;
- covering the data from different experiments or known clusters, e.g., from both liquid and gas phase when focusing on the EOS model, by each of the reduced datasets;
- sorting the data points according to decreasing weights in the objective function to account for data points with a high impact on the objective first;
- sorting the data points according to increasing measurement uncertainties to account for the most reliable data points first.

Apart from this, we may apply approaches from statistics, machine learning, and optimal experimental design to optimally choose the data points for augmenting. This ranges from adding the data points with the currently largest prediction error [compare 122] via choosing the next data points according to an acquisition function from Bayesian optimization [104, 119, 141, 182, 213] to picking data points which help to minimize the confidence region or maximum prediction error for the solution obtained based on the next larger dataset. However, the optimal choice of data points may be node-specific which would prevent us from re-using the reduced datasets. Overall, the computational effort for any of these more sophisticated approaches from statistics, machine learning, and optimal experimental design may easily exceed their benefit since we eventually converge to the full dataset provided with the model – no matter whether we choose the data points for augmentation randomly or optimally.

4. Conclusions and Outlook

Only concise mathematical models and efficient numerical programming methods allow for handling real-world optimization problems. We study the applicability of steady-state assumptions for reducing the complexity of dynamic models and propose a novel deterministic global optimization algorithm tailored to the general structure of parameter estimation problems. As both of these topics may be treated independently, we provided separate conclusions and future perspectives at the end of the respective chapters. In this joint closing chapter, we want to emphasize the synergies of the two topics once again.

Conclusions. Quasi-steady behavior is commonly assumed if the optimization of the operation is a lower-level task underlying a bigger research question, e.g., within the optimization of scheduling or design of energy supply systems. We have systematically studied how system-inherent inertia with a varying degree of time delay affects the optimal operation of an energy supply system. For this, we presented standard dynamic models extending the quasi-steady models of [209] and identified their behavior as first-order dynamics. Both theoretical and numerical investigations with a single boiler unit for fulfilling a realistic volatile heating demand showed that quasi-steady models are only adequate if the response of the boiler is fast compared to the chosen discretization of the demand profile. The presence of slow dynamics may prevent the energy system (component) from fulfilling the demand and, thus, requires the use of adequate ramping constraints when using steady-state assumptions. Conversely, we may underestimate the flexibility of the dynamic operation when assuming too strict ramping constraints. A subsequent case study on the optimal operation of a multi-component energy system for the integrated heating and electricity supply confirmed that the response time of dynamic components mainly effects the peak values of the control variables. However, even for fast dynamics and even in presence of a thermal energy storage, the optimal dynamic operational strategy varied significantly from the optimal quasi-steady operational strategy. Hence, the case study with the complex energy supply system concurs that adequate ramping constraints are required to obtain realistic operational strategies. Determining the corresponding ramps requires the solution of a parameter estimation problem if (a part of) the system's dynamics is only known from data rather than from a physical model.
Parameter estimation for fitting nonconvex models, e.g., nonlinear dynamics, to large measurement datasets results in a large-scale and, potentially, multi-modal optimization problem. This, in turn, requires efficient deterministic global optimization methods to guarantee the (global) optimality of the calculated solution. Thus, we proposed the B&B algorithm with growing datasets which combines the standard B&B algorithm with a data augmentation step. In detail, we start with a reduced dataset in the root node and let it grow by augmenting data points until reaching the full dataset provided originally with the model. In this way, we can save computational effort by bounding B&B nodes based on a reduced dataset at the beginning of the B&B algorithm, while still finding

the global solution of the original problem. We have proven that the proposed procedure gives indeed a deterministic global optimization method. Specifically for error-prone measurement datasets, the tightness of the lower bounds may suffer from the deterministic bounding based on reduced datasets. We therefore proposed two additional approaches applying heuristic lower bounds for pruning together with a post-processing procedure to detect and quantify potential mistakes made by the heuristic pruning. Our extensive benchmark study with 13 real-world parameter estimation problems suggests that even the heuristic approaches are almost deterministically converging towards the global solution, as indicated by the post-processing procedure, while providing significantly tighter lower bounds than the standard B&B algorithm within a given CPU time limit in most of the cases. As we draw the data points for initializing and augmenting the dataset randomly from the full dataset, we implemented a common resampling method from statistics. Our numerical results indicate that the B&B algorithm with growing datasets does not suffer from large variances.

Outlook. Combining the underlying challenges of the two main chapters of this thesis, brings us to parameter estimation subject to dynamic systems. When solving dynamic optimization problems, a crucial amount of computational effort is spent on integrating the dynamic system. When concluding the chapter on the B&B algorithm with growing datasets in Section 3.7, we proposed the use of ordered datasets. There are two apparent sorting orders for dynamic optimization problems which may also be combined. Firstly, we can sort the data points with time such that we stop the integration of the dynamic step at an earlier time point when using a reduced dataset. Secondly, we can cluster the data points, if applicable, by the different experimental conditions used. For example, we may evaluate the dynamic system only for one pair of input values when using the initial dataset, and add the measurements based on another pair of inputs when augmenting. Which of these sorting orders is more favorable depends on the structure of the dynamic system: Do we (expect to) have a stable dynamic system such that the optimal solution and, thus, the lower and upper bounds depend primarily on the first data points? Can we reuse function evaluations of intermediate variables for calculating the trajectories based on different pairs of input?
Furthermore, we can enhance the performance of the B&B algorithm with growing datasets for dynamic parameter estimation problems by applying approaches tailored to the deterministic global optimization of dynamic systems [e.g., 60, 108, 152, 187]. In particular, Diedam and Sager [51] show that multiple shooting is more favorable than single shooting for the global optimization of optimal control problems. Starting the integration of the dynamics from scratch in each shooting interval allows to limit the error of the relaxations accumulated over the time steps. Note that [51] have proven that we do not need to branch on the auxiliary optimization variables introduced for the initialization of the dynamic trajectory in each shooting interval, meaning that the number of unknown parameters may remain the same even when using multiple shooting. Choosing the measurement intervals as shooting intervals allows for an efficient model formulation [51, 52] and, also, for using the measurement data points as a start solution for the auxiliary shooting variables. A good start solution supports the efficient solution of both dynamic and global optimization problems.

Appendix A.

Simplified dynamic models for energy supply systems

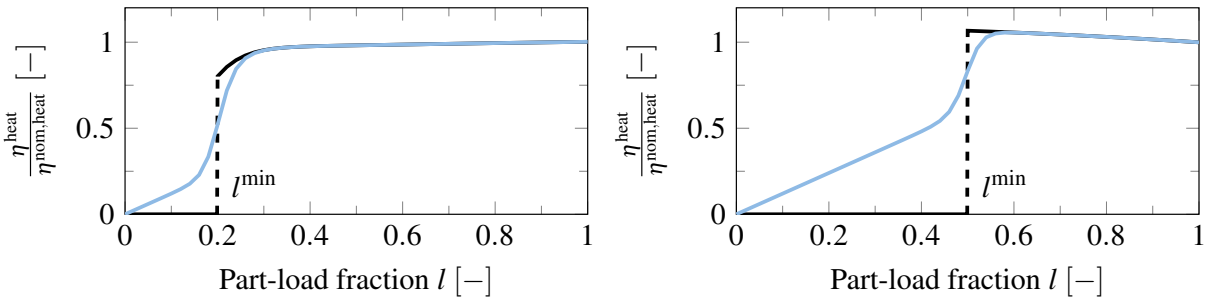
A.1. Energy system ES1 and building model

A.1.1. Energy supply system ES1

Based on Section 2.3, the dynamic optimization problem for the optimal operation of energy system ES1 depicted in Figure 2.2 is given by

$$\min_{\substack{\dot{Q}^{\mathrm{in}}_{\mathrm{BO}}(\cdot),\ \dot{Q}^{\mathrm{in}}_{\mathrm{CHP}}(\cdot),\\ P_{\mathrm{PV1}}(\cdot),\ P_{\mathrm{PV2}}(\cdot),\ P^{\mathrm{buy}}(\cdot),\\ l_{\mathrm{BO},0},\ l_{\mathrm{CHP},0}}} \int_0^{168} c^{\mathrm{gas}}(t)\cdot\left(\dot{Q}^{\mathrm{in}}_{\mathrm{BO}}(t)+\dot{Q}^{\mathrm{in}}_{\mathrm{CHP}}(t)\right) + c^{\mathrm{el,buy}}(t)\cdot P^{\mathrm{buy}}(t) - c^{\mathrm{el,sell}}(t)\cdot P^{\mathrm{sell}}(t)\ \mathrm{d}t$$

$$\begin{aligned}
\text{s.t.}\quad \frac{\mathrm{d}l_i}{\mathrm{d}t}(t) &= \frac{1}{\tau_i}\left(\hat{\eta}^{\mathrm{th}}_i(l_i(t))\cdot\frac{\dot{Q}^{\mathrm{in}}_i(t)}{\dot{Q}^{\mathrm{nom}}_i}-l_i(t)\right) && \forall i\in\{\mathrm{BO},\mathrm{CHP}\}\ \forall t\in[0,168]\\
l_i(0) &= l_{i,0} && \forall i\in\{\mathrm{BO},\mathrm{CHP}\}\\
\dot{Q}^{\mathrm{out}}_i(t) &= l_i(t)\cdot\dot{Q}^{\mathrm{nom}}_i && \forall i\in\{\mathrm{BO},\mathrm{CHP}\}\ \forall t\in[0,168]\\
\frac{\mathrm{d}E_{\mathrm{TES}}}{\mathrm{d}t}(t) &= \eta_{\mathrm{TES}}\cdot\dot{Q}_{\mathrm{TES}}(t)-\frac{1}{\tau^{\mathrm{loss}}_{\mathrm{TES}}}\cdot E_{\mathrm{TES}}(t) && \forall t\in[0,168]\\
E_{\mathrm{TES}}(0) &= \frac{1}{2}E^{\mathrm{nom}}_{\mathrm{TES}}\\
\dot{Q}_{\mathrm{TES}}(t) &= \dot{Q}^{\mathrm{out}}_{\mathrm{BO}}(t)+\dot{Q}^{\mathrm{out}}_{\mathrm{CHP}}(t)-\dot{Q}^{\mathrm{th,dem}}(t) && \forall t\in[0,168]\\
P_{\mathrm{CHP}}(t) &= \hat{\eta}^{\mathrm{el}}_{\mathrm{CHP}}(l_{\mathrm{CHP}}(t))\cdot\dot{Q}^{\mathrm{in}}_{\mathrm{CHP}}(t) && \forall t\in[0,168]\\
P^{\mathrm{sell}}(t) &= P_{\mathrm{PV1}}(t)+P_{\mathrm{PV2}}(t)+P^{\mathrm{buy}}(t)+P_{\mathrm{CHP}}(t)-P^{\mathrm{dem}}(t) && \forall t\in[0,168]\\
l_i(t) &\in[0,1] && \forall i\in\{\mathrm{BO},\mathrm{CHP}\}\ \forall t\in[0,168]\\
E_{\mathrm{TES}}(t) &\in[0,E^{\mathrm{nom}}_{\mathrm{TES}}] && \forall t\in[0,168]\\
\dot{Q}_{\mathrm{TES}}(t) &\in[-\dot{Q}^{\mathrm{nom}}_{\mathrm{TES}},\dot{Q}^{\mathrm{nom}}_{\mathrm{TES}}] && \forall t\in[0,168]\\
P^{\mathrm{sell}}(t) &\geq 0 && \forall t\in[0,168]\\
\dot{Q}^{\mathrm{in}}_i(t) &\geq 0 && \forall i\in\{\mathrm{BO},\mathrm{CHP}\}\ \forall t\in[0,168]\\
P_i(t) &\in[0,P^{\mathrm{irr}}_i(t)] && \forall i\in\{\mathrm{PV1},\mathrm{PV2}\}\ \forall t\in[0,168]\\
P^{\mathrm{buy}}(t) &\geq 0 && \forall t\in[0,168]
\end{aligned}$$



(a) Boiler **(b)** CHP

Figure A.1.: Smoothed relative efficiency with minimum part-load fraction $l^{\min}$

with a-priori given costs for purchasing gas c^{gas} and electricity $c^{\text{el,buy}}$ as well as for selling electricity $c^{\text{el,sell}}$. Note that the initial values for the part-load fraction of the boiler and CHP unit are optimization variables as well. Note further that control functions $\dot{Q}^{\text{in}}_{\text{BO}}(\cdot)$ and $\dot{Q}^{\text{in}}_{\text{CHP}}(\cdot)$ are bounded from above via the bound on the part-load fraction, control function $P^{\text{buy}}(\cdot)$ via cost optimality, and differential state $P^{\text{sell}}(\cdot)$ via the limited capacities of the system components as well as cost optimality.

In the model of energy system ES1, we include a minimum part-load fraction into the dynamic model equations of the boiler and CHP unit. More precisely, the hyperbolic tangent is used as a smooth switching function between turned off mode and an operation with a part-load fraction larger than the minimum part-load fraction. In this way, we can avoid adding a combinatorial complexity to a dynamic model since no binary controls are introduced.

As a complication, the turned off mode with part-load fraction $l = 0$ poses a stable point of the dynamics of the boiler and CHP unit. In fact, point $(\frac{\mathrm{d}l}{\mathrm{d}t}, l) = (0, 0)$ cannot be escaped independent of the chosen control value $\dot{Q}^{\text{in}}_i$

$$\begin{aligned}\frac{\mathrm{d}l}{\mathrm{d}t}(t)\bigg|_{l(t)=0} &= \frac{1}{\tau_i} \cdot \left(\eta^{\text{th}}_i(l(t))\big|_{l(t)=0} \cdot \frac{\dot{Q}^{\text{in}}_i}{\dot{Q}^{\text{nom}}_i} - 0 \right) \\ &= \frac{1}{\tau_i} \cdot \left(0 \cdot \frac{\dot{Q}^{\text{in}}_i}{\dot{Q}^{\text{nom}}_i} - 0 \right) = 0 \qquad \forall\, i \in \{\text{BO}, \text{CHP}\}\end{aligned}$$

without a binary control turning the boiler or CHP unit explicitly on. Therefore, the zero-operation below the minimum part-load fraction is replaced by a linear operation with sufficient slope, see Figure A.1. Consequently, we replace original efficiency $\eta^{\text{th}}_i(l)$, $i \in \{\text{BO}, \text{CHP}\}$ by the adapted smooth efficiency

$$\begin{aligned}\hat{\eta}^{\text{th}}_i(l) := &\left(0.5 + 0.5 \cdot \tanh\left(a_1 \cdot (l - l^{\min}_i)\right)\right) \cdot \eta^{\text{th}}_i(l) \\ &+ \left(0.5 - 0.5 \cdot \tanh\left(a_1 \cdot (l - l^{\min}_i)\right)\right) \cdot (a_2 + a_3 \cdot l)\ . \end{aligned} \tag{A.1}$$

Since the use of the auxiliary linear operation in (A.1) already allows the CHP unit to escape stable point $(\frac{\mathrm{d}l}{\mathrm{d}t}, l) = (0, 0)$, we can use a zero-operation below the minimum part-load fraction for the electric efficiency

$$\hat{\eta}^{\text{el}}_{\text{CHP}}(l) := \left(0.5 + 0.5 \cdot \tanh\left(a_1 \cdot (l - l^{\min}_{\text{CHP}})\right)\right) \cdot \eta^{\text{th}}_{\text{CHP}}(l)\ .$$

As an example, parameter values $a_1 = 30$, $a_2 = 0.0001$, and $a_3 = 1.2$ are chosen.
The underlying efficiencies of the CHP units

$$\eta_{\mathrm{CHP}}^{\mathrm{th}}(l) = \left(-0.0768 \cdot l^2 - 0.0199 \cdot l + 1.0960\right) \cdot \eta_{\mathrm{CHP}}^{\mathrm{nom,th}}(\dot{Q}_{\mathrm{CHP}}^{\mathrm{nom}}) \ ,$$
$$\eta_{\mathrm{CHP}}^{\mathrm{el}}(l) = \left(-0.2611 \cdot l^2 + 0.6743 \cdot l + 0.5868\right) \cdot \eta_{\mathrm{CHP}}^{\mathrm{nom,el}}(\dot{Q}_{\mathrm{CHP}}^{\mathrm{nom}})$$

are approximated to information provided by the manufacturer, while the efficiency of the boiler

$$\eta_{\mathrm{BO}}^{\mathrm{th}}(l) = \frac{21.75378 \cdot l^3 - 7.00130 \cdot l^2 + 1.39731 \cdot l - 0.07557}{20.66646 \cdot l^3 - 5.34196 \cdot l^2 + 0.67774 \cdot l + 0.03487} \cdot \eta_{\mathrm{BO}}^{\mathrm{nom,th}}$$

is given by [207] based on [61]. We use the nominal efficiencies from [207]

$$\eta_{\mathrm{CHP}}^{\mathrm{nom,th}}(\dot{Q}_{\mathrm{CHP}}^{\mathrm{nom}}) = 0.498 - 3.55 \cdot 10^{-5} \cdot \frac{\dot{Q}_{\mathrm{CHP}}^{\mathrm{nom}}}{1\,\mathrm{kW}} \ , \tag{A.2}$$

$$\eta_{\mathrm{CHP}}^{\mathrm{nom,el}}(\dot{Q}_{\mathrm{CHP}}^{\mathrm{nom}}) = 0.372 + 3.55 \cdot 10^{-5} \cdot \frac{\dot{Q}_{\mathrm{CHP}}^{\mathrm{nom}}}{1\,\mathrm{kW}} \ , \tag{A.3}$$

$$\eta_{\mathrm{BO}}^{\mathrm{nom,th}} = 0.8 \ .$$

A.1.2. Building model

We model the inner temperature T within a building with one thermal zone by

$$\frac{\mathrm{d}T}{\mathrm{d}t}(t) = \frac{1}{C} \cdot \left(\gamma^{\mathrm{amb}} \cdot \left(T^{\mathrm{amb}}(t) - T(t)\right) + \gamma^{\mathrm{irr}} \cdot P^{\mathrm{irr}}(t) + \gamma^{\mathrm{th}} \cdot \dot{Q}^{\mathrm{dem}}(t)\right) \qquad \forall t \in [0, 168]$$
$$T(0) = 294.15$$

with constants $C = 1\,530.4\,\mathrm{MW\,s/K}$, $\gamma^{\mathrm{amb}} = 3\,651\,\mathrm{W/K}$, $\gamma^{\mathrm{irr}} = 0.037$, and $\gamma^{\mathrm{th}} = 1$ as proposed by Dominique Sauer[1], see building "OB 1" in Appendix B2 of [170]. The time series for ambient temperature T^{amb}, utilizable solar irradiation P^{irr}, and heating demand $\dot{Q}^{\mathrm{dem}}$ are provided a-priori.

[1] Affiliation: Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, 76021 Karlsruhe, Germany

A.2. Energy supply system ES2

A.2.1. Dynamic optimization problem

Based on Section 2.3, the dynamic model of energy supply system ES2 comprising components $\mathcal{C} = \{\mathrm{BO}, \mathrm{CHP1}, \mathrm{CHP2}\}$ and a TES unit as depicted in Figure 2.7 reads as

$$\min_{\substack{\dot{Q}_i^{\mathrm{in}}(\cdot),\ l_{i,0}, i\in\mathcal{C},\\ \dot{Q}_{\mathrm{TES}}(\cdot),\ E_{\mathrm{TES},0},\\ P^{\mathrm{buy}}(\cdot),\ P^{\mathrm{sell}}(\cdot)}} \int_0^{168} c^{\mathrm{gas}}(t)\cdot\sum_{i\in\mathcal{C}}\dot{Q}_i^{\mathrm{in}}(t) + c^{\mathrm{el,buy}}(t)\cdot P^{\mathrm{buy}}(t) - c^{\mathrm{el,sell}}(t)\cdot P^{\mathrm{sell}}(t)\ \mathrm{d}t$$

$$\begin{aligned}
\text{s.t.}\quad \frac{\mathrm{d}l_i}{\mathrm{d}t}(t) &= \frac{1}{\tau_i}\left(\eta_i^{\mathrm{th}}(l_i(t))\cdot\frac{\dot{Q}_i^{\mathrm{in}}(t)}{\dot{Q}_i^{\mathrm{nom}}} - l_i(t)\right) && \forall i\in\mathcal{C}\ \forall t\in[0,168]\\
l_i(0) &= l_{0,i} && \forall i\in\mathcal{C}\\
\dot{Q}_i^{\mathrm{out}}(t) &= l_i(t)\cdot\dot{Q}_i^{\mathrm{nom}} && \forall i\in\mathcal{C}\ \forall t\in[0,168]\\
\frac{\mathrm{d}E_{\mathrm{TES}}}{\mathrm{d}t}(t) &= \dot{Q}_{\mathrm{TES}}(t) - \frac{1}{\tau_{\mathrm{TES}}^{\mathrm{loss}}}\cdot E_{\mathrm{TES}}(t) && \forall t\in[0,168]\\
E_{\mathrm{TES}}(0) &= E_{\mathrm{TES},0}\\
\dot{Q}^{\mathrm{th,dem}}(t) &= \sum_{i\in\mathcal{C}}\dot{Q}_i^{\mathrm{out}}(t) - \dot{Q}_{\mathrm{TES}}(t) && \forall t\in[0,168]\\
P_i(t) &= \eta_i^{\mathrm{el}}(l_i(t))\cdot\dot{Q}_i^{\mathrm{in}}(t) && \forall i\in\{\mathrm{CHP1},\mathrm{CHP2}\}\ \forall t\in[0,168]\\
P^{\mathrm{dem}}(t) &= P^{\mathrm{buy}}(t) + P_{\mathrm{CHP1}}(t) + P_{\mathrm{CHP2}}(t) - P^{\mathrm{sell}}(t) && \forall t\in[0,168]\\
l_i(t) &\in [0,1] && \forall i\in\mathcal{C}\ \forall t\in[0,168]\\
l_{\mathrm{CHP1}}(t) &\leq l_{\mathrm{CHP2}}(t) && \forall t\in[0,168]\\
E_{\mathrm{TES}}(t) &\in [0, E_{\mathrm{TES}}^{\mathrm{nom}}] && \forall t\in[0,168]\\
\dot{Q}_i^{\mathrm{in}}(t) &\geq 0 && \forall i\in\mathcal{C}\ \forall t\in[0,168]\\
\dot{Q}_{\mathrm{TES}}(t) &\in [-\frac{E_{\mathrm{TES}}(t)}{\tau_{\mathrm{TES}}}, \frac{E_{\mathrm{TES}}^{\mathrm{nom}} - E_{\mathrm{TES}}(t)}{\tau_{\mathrm{TES}}}] && \forall t\in[0,168]\\
P^{\mathrm{sell}}(t) &\geq 0 && \forall t\in[0,168]\\
P^{\mathrm{buy}}(t) &\geq 0 && \forall t\in[0,168]
\end{aligned}$$

with a-priori given costs for purchasing gas c^{gas} and electricity $c^{\mathrm{el,buy}}$ as well as for selling electricity $c^{\mathrm{el,sell}}$. Note that the heating demand needs to be strictly satisfied, while left over electricity is sold. Note further that we impose a symmetry breaking constraint on the part-load fractions of the CHP units to avoid an unnecessary expansion of the feasible region of operational strategies. In contrast to the model for ES1, see Appendix A.1.1, we do not account for discharging and charging losses, i.e., we fix $\eta_{\mathrm{TES}} = 1$, and bound the heat transfer rate exchanged with the TES explicitly by the currently available storage level. In accordance with the model for ES1, control functions $\dot{Q}_i^{\mathrm{in}},\ i\in\mathcal{C}$ are bounded from above via the bound on the part-load fraction, and the power flow indicating sold and purchased electricity are bounded due to cost optimality and the limited capacities of the system components.

We use polynomial curves

$$\begin{aligned}
\eta_{\mathrm{BO}}^{\mathrm{th}}(l_{\mathrm{BO}}) =& \frac{2.0230 \cdot l_{\mathrm{BO}}^3 + 0.3439 \cdot l_{\mathrm{BO}}^2 + 0.5634 \cdot l_{\mathrm{BO}}}{l_{\mathrm{BO}}^3 + 1.7500 \cdot l_{\mathrm{BO}}^2 + 0.0672 \cdot l_{\mathrm{BO}} + 0.1122} \cdot \eta_{\mathrm{BO}}^{\mathrm{nom,th}} \ , \\
\eta_i^{\mathrm{th}}(l_i) =& (-0.2667 \cdot l_i^2 + 0.5053 \cdot l_i + 0.7605) \cdot \eta_{\mathrm{CHP}}^{\mathrm{nom,th}}(\dot{Q}_{\mathrm{CHP}}^{\mathrm{nom}}) \quad i \in \{\mathrm{CHP1}, \mathrm{CHP2}\} \ , \\
\eta_i^{\mathrm{el}}(l_i) =& \left(-0.5533 \cdot l_i^2 + 1.4660 \cdot l_i + 0.0885\right) \cdot \eta_{\mathrm{CHP}}^{\mathrm{nom,el}}(\dot{Q}_{\mathrm{CHP}}^{\mathrm{nom}}) \quad i \in \{\mathrm{CHP1}, \mathrm{CHP2}\} \ ,
\end{aligned}$$

fitted to the relative efficiencies depicted in Figure A.2 of [207]. For the nominal efficiencies, we use (A.2) and (A.3) from [207] as well as $\eta_{\mathrm{BO}}^{\mathrm{nom,th}} = 0.9$. In particular, we do not account for minimum part-load fractions for ES2.

A.2.2. Quasi-steady optimization problem

To obtain a quasi-steady model, all continuous-time trajectories of the model presented in Section A.2.1 are replaced with time series holding values for time points $0 = t_0 < t_1 < \ldots < t_{n_t} = 168$ and the time derivatives of the part-load fractions are set to zero. Since storage models are intrinsically dynamic, we follow the standard approach to discretize the continuous-time dynamics using the explicit Euler scheme [18, 64, 180] for adapting the TES model to the algebraic optimization framework of the quasi-steady formulations. The quasi-steady model of ES2 is given by

$$\min_{\substack{\dot{Q}_i^{\mathrm{in}}(t_j), i \in \mathcal{C}, \\ \dot{Q}_{\mathrm{TES}}(t_j),\ E_{\mathrm{TES}}(t_0), \\ P^{\mathrm{buy}}(t_j),\ P^{\mathrm{sell}}(t_j) \\ \forall j = 0, \ldots, 167}} \sum_{j=0}^{167} \Delta t \cdot \Bigg(c^{\mathrm{gas}}(t_j) \cdot \sum_{i \in \mathcal{C}} \dot{Q}_i^{\mathrm{in}}(t_j) + c^{\mathrm{el,buy}}(t_j) \cdot P^{\mathrm{buy}}(t_j) - c^{\mathrm{el,sell}}(t_j) \cdot P^{\mathrm{sell}}(t_j) \Bigg)$$

$$\begin{aligned}
\text{s.t.} \quad l_i(t_j) &= \eta_i^{\mathrm{th}}(l_i(t_j)) \cdot \frac{\dot{Q}_i^{\mathrm{in}}(t_j)}{\dot{Q}_i^{\mathrm{nom}}} && \forall i \in \mathcal{C}\ \forall j \in \{0, 167\} \\
\dot{Q}_i^{\mathrm{out}}(t_j) &= l_i(t_j) \cdot \dot{Q}_i^{\mathrm{nom}} && \forall i \in \mathcal{C}\ \forall j \in \{0, 167\} \\
\frac{E_{\mathrm{TES}}(t_{j+1}) - E_{\mathrm{TES}}(t_j)}{t_{j+1} - t_j} &= \dot{Q}_{\mathrm{TES}}(t_j) - \frac{1}{\tau_{\mathrm{TES}}^{\mathrm{loss}}} \cdot E_{\mathrm{TES}}(t_j) && \forall j \in \{0, 167\} \\
\dot{Q}^{\mathrm{th,dem}}(t_j) &= \sum_{i \in \mathcal{C}} \dot{Q}_i^{\mathrm{out}}(t_j) - \dot{Q}_{\mathrm{TES}}(t_j) && \forall j \in \{0, 167\} \\
P_i(t_j) &= \eta_i^{\mathrm{el}}(l_i(t_j)) \cdot \dot{Q}_i^{\mathrm{in}}(t_j) && \forall i \in \{\mathrm{CHP1}, \mathrm{CHP2}\}\ \forall j \in \{0, 167\} \\
P^{\mathrm{dem}}(t_j) &= P^{\mathrm{buy}}(t_j) + P_{\mathrm{CHP1}}(t_j) + P_{\mathrm{CHP2}}(t_j) - P^{\mathrm{sell}}(t_j) && \forall j \in \{0, 167\} \\
l_i(t_j) &\in [0, 1] && \forall i \in \mathcal{C}\ \forall j \in \{0, 167\} \\
l_{\mathrm{CHP1}}(t_j) &\leq l_{\mathrm{CHP2}}(t_j) && \forall j \in \{0, 167\} \\
E_{\mathrm{TES}}(t_j) &\in [0, E_{\mathrm{TES}}^{\mathrm{nom}}] && \forall j \in \{0, 167\} \\
\dot{Q}_i^{\mathrm{in}}(t_j) &\geq 0 && \forall i \in \mathcal{C}\ \forall j \in \{0, 167\} \\
\dot{Q}_{\mathrm{TES}}(t_j) &\in [-\frac{E_{\mathrm{TES}}(t_j)}{\tau_{\mathrm{TES}}}, \frac{E_{\mathrm{TES}}^{\mathrm{nom}} - E_{\mathrm{TES}}(t_j)}{\tau_{\mathrm{TES}}}] && \forall j \in \{0, 167\} \\
P^{\mathrm{sell}}(t_j) &\geq 0 && \forall j \in \{0, 167\} \\
P^{\mathrm{buy}}(t_j) &\geq 0 && \forall j \in \{0, 167\}
\end{aligned}$$

with constant step width $\Delta t := t_{j+1} - t_j = 1\,\mathrm{h}$, $j = 0, \ldots, 167$. Note that we consider an hourly discretization for a time horizon of 2 weeks starting with time point $t_0 = 0$ and ending with time point $t_{n_t-1} = 167\,\mathrm{h}$. This is consistent with the dynamic model where the last time point $t_f = 168\,\mathrm{h}$ has no volume to consider in the integral over the energy costs. Note further that we gain the quasi-steady models of Voll et al. (2013) [209] for the boiler and CHP units except that we do not enforce a minimum part-load fraction larger than zero. In particular, the part-load fractions are determined by the respective input heat transfer rate for any time point $t_j,\ j \geq 0$.

Appendix B.

Further information on the global optimization of reduced problems

B.1. Parameter points chosen for function evaluations

Table B.1.: Parameter points giving objective values with different orders of magnitude for problem EOS262

Point	n_{10}	t_{10}	d_{10}	c_{10}	l_{10}	η_{10}	β_{10}	γ_{10}	ε_{10}
1	-0.604	3.25	2	1	2	0.0	0.0	0.0	0.0
2	-0.640	2.00	2	1	2	1.0	1.0	2.0	1.5
3	2.000	22.00	10	0	1	0.0	0.0	2.0	1.5

B.2. Minimizing the mean squared error

The Figures B.1 and B.2 as well as Table B.2 are obtained when repeating the study in Section 3.3 using the mean squared error (MSE) rather than the summed squared error (SSE) as the objective function.

Table B.2.: Order of magnitude of final lower and upper bound for the base scenario and sample scenarios 50_1 to 50_5 and 100_1 to 100_5 indicated by the minimum value, maximum value, and median when dividing the parameter boxes into two categories

	Base scenario			Sample scenarios		
	min	max	median	min	max	median
Final lower bound						
good regions[a]	0.00	$1.71 \cdot 10^{-4}$	$1.12 \cdot 10^{-4}$	0.00	$6.68 \cdot 10^{-3}$	$2.77 \cdot 10^{-5}$
bad regions[b]	$1.66 \cdot 10^{2}$	$5.38 \cdot 10^{14}$	$8.30 \cdot 10^{5}$	$8.93 \cdot 10^{-21}$	$1.31 \cdot 10^{14}$	$2.61 \cdot 10^{3}$
Final upper bound						
good regions[a]	$5.67 \cdot 10^{-8}$	$8.30 \cdot 10^{5}$	$1.31 \cdot 10^{-1}$	$1.27 \cdot 10^{-18}$	$5.01 \cdot 10^{3}$	$2.08 \cdot 10^{-3}$
bad regions[b]	$1.72 \cdot 10^{2}$	$5.38 \cdot 10^{14}$	$8.30 \cdot 10^{5}$	$1.39 \cdot 10^{-4}$	$1.31 \cdot 10^{14}$	$2.61 \cdot 10^{3}$

[a]Set of all boxes with lower bound < 0.1 for the base scenario, i.e., boxes which potentially contain a perfect fit

[b]Set of all boxes with lower bound ≥ 0.1 for the base scenario, i.e., boxes which cannot contain a perfect fit

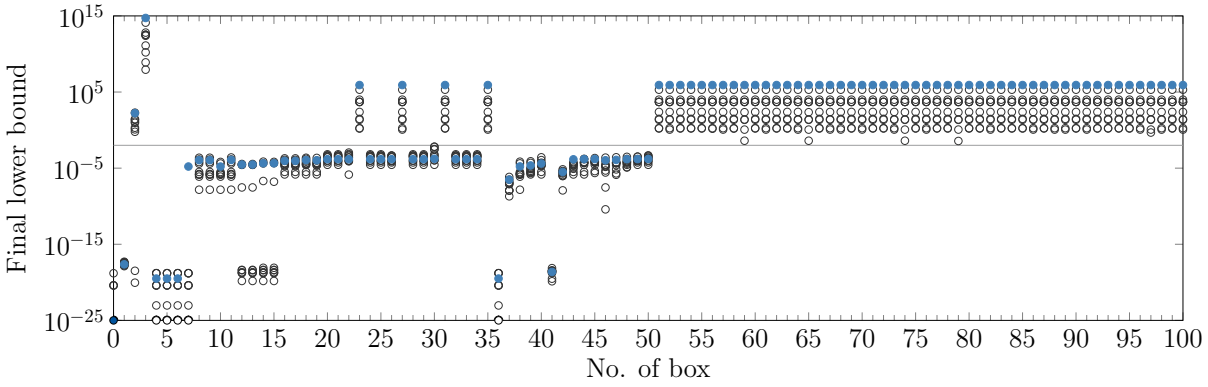


Figure B.1.: Final lower bound obtained for problem EOS262 with the standard B&B algorithm for the base scenario (•) as well as sample scenarios 50_1 to 50_5 and 100_1 to 100_5 (o) when optimizing over different parameter boxes, where Box 0 is the whole feasible region, with an optimality tolerance of 0.01 (–). Marks on the x-axis indicate values equal to 0.

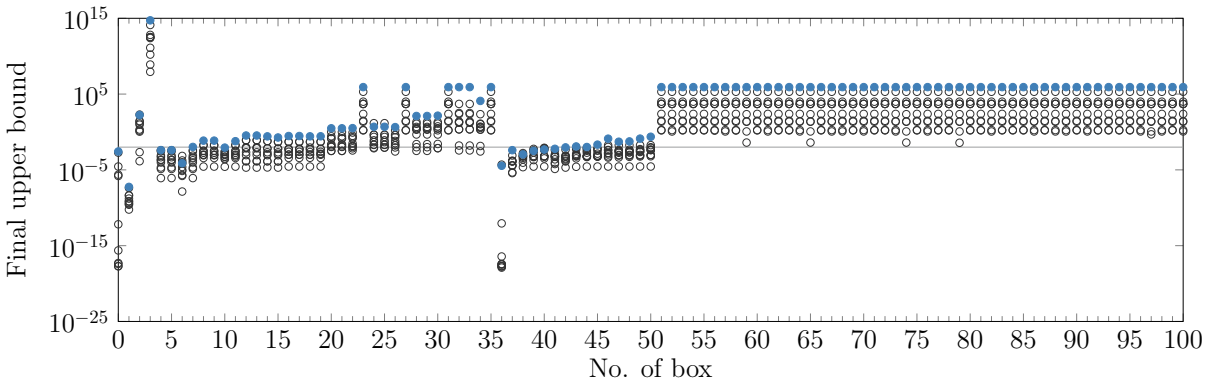


Figure B.2.: Final upper bound obtained for problem EOS262 with the standard B&B algorithm for the base scenario (•) as well as sample scenarios 50_1 to 50_5 and 100_1 to 100_5 (o) when optimizing over different parameter boxes, where Box 0 is the whole feasible region, with an optimality tolerance of 0.01 (–)

Appendix C.

Basic definitions used in the proof of convergence

In this section, we repeat the definitions used by Locatelli and Schoen [131] in the notation of this thesis. For this, we denote the feasible set of the node $N_k = (\mathcal{P}_k, \mathcal{D}_k)$ with associated parameter domain $\mathcal{P}_k \subsetneq \mathbb{R}^n$ and dataset $\mathcal{D}_k \subsetneq \mathbb{R}^m \times \mathbb{R}$ processed in B&B iteration k by $\mathcal{F}_k := \mathcal{P}_k \cap \left\{\boldsymbol{p} \in \mathcal{P} :\ h(\boldsymbol{x}_d, y_d; \boldsymbol{p}) \leq 0\ \forall (\boldsymbol{x}_d, y_d) \in \mathcal{D}\ \wedge\ \widetilde{h}(\boldsymbol{p}) \leq 0\right\}$, where $h(\boldsymbol{x}_d, y_d;)$ and $\widetilde{h}$ are the residuals of inequality constraints.

Definition 2 (Equation (5.24) of [131])**.** An underestimator $F^{\text{cv}} : \mathcal{P} \to \mathbb{R}$ satisfies the *isotonic property* if

$$\mathcal{F}_l \subseteq \mathcal{F}_k \quad \Rightarrow \quad F^{\text{cv}}(\boldsymbol{p})\big|_{\boldsymbol{p}\in\mathcal{F}_l} \geq F^{\text{cv}}(\boldsymbol{p})\big|_{\boldsymbol{p}\in\mathcal{F}_k} \quad \forall \boldsymbol{p} \in \mathcal{F}_l\ .$$

Definition 3 (Definition 5.4 of [131])**.** A convex underestimator $F^{\text{cv}} : \mathcal{P} \to \mathbb{R}$ of a function F over some region $\mathcal{F}_k$ is *exact in the limit* if

$$\max_{\boldsymbol{p}\in\mathcal{F}_k}\{F(\boldsymbol{p}) - F^{\text{cv}}(\boldsymbol{p})\} \leq \eta(\text{diam}(\mathcal{F}_k))\ ,$$

where $\text{diam}(\mathcal{F}_k) = \max_{\boldsymbol{p}_1,\boldsymbol{p}_2\in\mathcal{F}_k} ||\boldsymbol{p}_1 - \boldsymbol{p}_2||_2$ is the diameter of $\mathcal{F}_k$, and $\eta(r)$ is a continuous nondecreasing function such that $\lim_{r\to 0}\eta(r) = 0$.

Definition 4 (Definition 5.5 of [131])**.** A B&B algorithm equipped with a geometric branching rule possesses the *exhaustiveness* property if each infinite nested sequence generated by the algorithm $(\mathcal{F}_{k_j})_j :\ \mathcal{F}_{k_0} = \mathcal{F}_0,\ k_j < k_{j+1},\ \mathcal{F}_{k_{j+1}} \subseteq \mathcal{F}_{k_j}\ \forall j = 0, 1, \ldots$ converges to a singleton $\text{diam}(\mathcal{F}_{k_j}) \to 0$ as $j \to \infty$.

Appendix D.

Further details on benchmarking the B&B algorithm with growing datasets

D.1. MAiNGO settings

Table D.1 lists the MAiNGO settings deviating from the default values used for both the standard B&B algorithm and the B&B algorithm with growing datasets in the benchmark study discussed in Sections 3.6.2 and 3.6.3.

Table D.1.: MAiNGO settings used for benchmarking the B&B algorithm with growing datasets

Name of the setting	Value	Description
epsilonA	$\frac{0.01}{\lvert\mathcal{D}\rvert}$	Absolute optimality tolerance
epsilonR	0.1	Relative optimality tolerance
maxTime	23 h	CPU time limit
BAB_alwaysSolveObbt	True[a]/False[b]	Use optimality based bound tightening?
LBP_solver	CLP[a]/Interval extensions[b]	Solver/approach used for the lower bounding problem
UBP_maxStepsPreprocessing	300	Maximum number of steps performed in the local optimization for pre-processing the root node
UBP_maxTimePreprocessing	600	CPU time limit of the local optimization for pre-processing the root node
UBP_solverBab	LBFGS	Solver for the upper bounding problem
UBP_maxStepsBab	10	Maximum number of steps performed for solving the upper bounding problem
UBP_maxTimeBab	20	CPU time limit for solving the upper bounding problem

[a]For models IHM, EIS, TSP, GMM, and trainANN

[b]For models EOS and kinetics

D.2. Values of final lower and upper bounds

Tables D.2 and D.3 list the final lower and upper bounds calculated by the standard B&B algorithm with different objective functions and the different approaches of the B&B algorithm with growing datasets for the 13 estimation problems listed in Table 3.5. We report the median values of 5 repetitive runs for each case. If applicable, the final lower bounds are adapted by the post-processing procedure proposed in Section 3.5.1.

Table D.2.: Final lower bound of the estimation problems investigated in the benchmark study calculated within a CPU time limit of 23 h with the different algorithmic approaches: the standard B&B algorithm with objective (SSE) and (MSE) as well as the deterministic and heuristic approaches of the B&B algorithm with growing datasets applying different augmentation rules. Bold numbers indicate the largest value for the respective problem.

	Standard B&B		Deterministic Appr. (SSE)			SSE heuristic			MSE[a] heuristic		
Problem	(SSE)	(MSE)[a]	CONST	SCALING	COMBI	CONST	SCALING	TOL	CONST	OOS	TOL
Without resampling											
EOS262	6.0e-16	6.0e-16	3.9e-16	1.5e-17	1.5e-17	**6.7e-16**	8.2e-18[b]	8.2e-18[b]	6.0e-16	6.5e-16	6.0e-16
EOS2262	6.1e-15	6.2e-15	5.2e-15	3.8e-16	3.8e-16	8.4e-15	**2.2e-14**	**2.2e-14**	4.7e-16	6.3e-15	6.4e-16
EOS262noisy	2.3e-02	2.5e-02	2.9e-03	1.8e-05	1.8e-05	5.7e-03	**2.9e-02**	2.7e-02	5.4e-03	5.9e-03	5.9e-03
EOS2262noisy	2.6e-03	2.9e-03	1.6e-03	1.8e-04	1.8e-04	2.9e-02	**5.2e-02**	5.1e-02	1.9e-03	2.2e-03	2.2e-03
IHMcon	1.2e+05	**1.3e+05**	6.6e+04	1.4e+04	1.7e+04	0	0	0	0	0	0
IHMunc	0	0	0	0	0	4.9e+03	**1.3e+06**	5.8e+04	4.2e+02	1.6e+05	1.6e+05
kinetics	0	2.6e-01	0	0	0	3.7e+00	**2.7e+01**	**2.7e+01**	3.7e+00	4.1e+00	4.1e+00
EIS	**9.8e+00**	**9.8e+00**	3.0e+00	0	0	2.4e+00	7.4e+00	8.2e+00	8.4e+00	4.5e+00	4.8e+00
TSPexact	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
TSPnoisy	1.6e-05	1.6e-05	1.6e-05	0	0	2.2e-04	**4.3e-04**	4.7e-05	1.8e-05	1.8e-05	2.4e-05
GMMcon	1.3e+02	1.4e+02	9.1e+01	1.4e+01	1.4e+01	2.0e+02	**4.6e+02**	4.1e+02[b]	1.1e+02	1.1e+02	1.1e+02
GMMineq	1.5e+02	1.4e+02	8.9e+01	1.6e+01	1.6e+01	3.1e+02	**5.4e+02**	2.0e+02[b]	1.5e+02	1.6e+02	1.7e+02
trainANN	6.2e-04	6.3e-04	6.2e-04	6.3e-05	6.3e-05	6.2e-04	**3.4e-03**	**3.4e-03**	5.1e-04	6.3e-04	6.3e-04
With resampling											
EOS262	6.0e-16	6.0e-16	3.9e-16	1.5e-17	1.5e-17	**6.7e-16**	8.2e-18[b]	8.2e-18[b]	6.0e-16	6.5e-16	6.0e-16
EOS2262	6.1e-15	6.2e-15	5.2e-15	5.3e-16	5.3e-16	8.3e-15	2.6e-14	**2.8e-14**	4.6e-16	6.4e-15	5.9e-16
EOS262noisy	2.3e-02	2.5e-02	1.8e-03	1.8e-05	1.8e-05	5.5e-03	**2.9e-02**	2.8e-02	5.3e-03	5.9e-03	5.9e-03
EOS2262noisy	2.6e-03	2.9e-03	1.5e-03	1.8e-04	1.8e-04	2.9e-02	**5.3e-02**	**5.3e-02**	1.8e-03	2.1e-03	2.1e-03
IHMcon	1.2e+05	**1.3e+05**	5.2e+04	5.3e+04	5.5e+04	0	0	0	0	0	0
IHMunc	0	0	0	0	0	3.3e+03	**1.2e+06**	2.8e+04	2.8e+00	1.5e+05	1.5e+05
kinetics	0	2.6e-01	0	0	0	3.7e+00	**2.7e+01**	2.6e+01	3.7e+00	4.1e+00	4.1e+00
EIS	**9.8e+00**	**9.8e+00**	2.9e+00	3.2e-04	2.2e-04	3.8e+00	9.0e+00	9.3e+00	8.1e+00	4.7e+00	4.3e+00
TSPexact	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
TSPnoisy	1.6e-05	1.6e-05	3.5e-07	3.2e-13	3.2e-13	2.2e-04	**4.1e-04**	4.7e-05	1.6e-05	1.8e-05	2.4e-05
GMMcon	1.3e+02	1.4e+02	8.9e+01	1.4e+01	1.4e+01	2.0e+02	**4.7e+02**	4.1e+02[b]	1.1e+02	1.1e+02	1.1e+02
GMMineq	1.5e+02	1.4e+02	9.4e+01	1.6e+01	1.6e+01	3.0e+02	**5.2e+02**	1.5e+02[b]	1.5e+02	1.6e+02	1.6e+02
trainANN	6.2e-04	6.3e-04	6.2e-04	7.8e-05	7.8e-05	6.2e-04	**5.8e-03**	5.6e-03	5.1e-04	7.8e-04	6.3e-04

[a]We multiplied the final lower bound calculated with objective (MSE) with $|\mathcal{D}|$ to enable a comparison to the methods using objective (SSE).

[b]Final lower bound decreased during post-processing.

Table D.3.: Best solution value found for the investigated estimation problems within a CPU time limit of 23 h with the different algorithmic approaches: the standard B&B algorithm with objective (SSE) and (MSE) as well as the deterministic and heuristic approaches of the B&B algorithm with growing datasets applying different augmentation rules. Bold numbers indicate the smallest value for the respective problem.

	Standard B&B		Deterministic Appr. (SSE)			SSE heuristic			MSE[a] heuristic		
Problem	(SSE)	(MSE)[a]	CONST	SCALING	COMBI	CONST	SCALING	TOL	CONST	OOS	TOL
Without resampling											
EOS262	6.3e-03	5.1e-03	8.9e-03	8.3e-03	5.6e-03	5.0e-03	1.9e+00	1.9e+00	**1.2e-03**	1.1e+01	8.9e-03
EOS2262	1.5e-01	1.8e-02	1.2e-01	**8.0e-03**	**8.0e-03**	1.3e+02	1.2e+02	1.2e+02	5.1e+01	1.2e+02	4.9e+00
EOS262noisy	8.2e+05	8.2e+05	**7.9e+05**	8.8e+05	8.8e+05	8.7e+05	9.2e+05	9.2e+05	9.2e+05	9.2e+05	9.2e+05
EOS2262noisy	**7.9e+07**	1.0e+08	1.1e+08	2.1e+08	2.1e+08	9.6e+07	1.8e+08	1.8e+08	1.0e+08	2.2e+08	2.2e+08
IHMcon	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**
IHMunc	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**
kinetics	**5.0e+02**	**5.0e+02**	5.3e+02	1.9e+04	9.0e+02	**5.0e+02**	5.4e+02	5.4e+02	5.1e+02	5.4e+02	5.4e+02
EIS	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**
TSPexact	9.1e-03	1.5e-02	2.5e-02	9.9e-03	2.5e-02	1.2e-02	1.2e-02	8.6e-03	**6.4e-03**	**6.4e-03**	**6.4e-03**
TSPnoisy	1.2e-01	2.3e-02	4.0e-02	4.0e-02	4.0e-02	2.7e-02	2.6e-02	2.2e-02	**2.1e-02**	**2.1e-02**	**2.1e-02**
GMMcon	1.1e+03	1.1e+03	1.1e+03	1.1e+03	1.1e+03	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	1.1e+03	1.1e+03
GMMineq	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**
trainANN	**7.2e+00**	7.4e+00	**7.2e+00**	**7.2e+00**	**7.2e+00**	**7.2e+00**	**7.2e+00**	**7.2e+00**	7.5e+00	7.7e+00	7.5e+00
With resampling											
EOS262	6.3e-03	5.1e-03	9.0e-03	9.2e-03	4.7e-03	3.0e-03	1.9e+00	1.9e+00	**1.2e-03**	1.1e+01	8.2e-03
EOS2262	1.5e-01	1.8e-02	1.2e-01	6.0e-03	**3.6e-03**	1.3e+02	1.2e+02	1.2e+02	5.1e+01	1.2e+02	4.9e+00
EOS262noisy	8.2e+05	8.2e+05	**8.1e+05**	8.5e+05	8.5e+05	8.7e+05	9.2e+05	9.2e+05	8.5e+05	9.2e+05	9.2e+05
EOS2262noisy	**7.9e+07**	1.0e+08	1.1e+08	2.1e+08	2.1e+08	9.6e+07	1.8e+08	1.8e+08	1.1e+08	2.2e+08	2.2e+08
IHMcon	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**
IHMunc	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**	**1.4e+07**
kinetics	**5.0e+02**	**5.0e+02**	5.3e+02	1.9e+04	1.5e+03	**5.0e+02**	5.4e+02	5.4e+02	5.1e+02	5.4e+02	5.4e+02
EIS	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**	**1.1e+01**
TSPexact	9.1e-03	1.5e-02	2.5e-02	9.9e-03	2.5e-02	1.2e-02	1.2e-02	8.6e-03	**6.4e-03**	**6.4e-03**	**6.4e-03**
TSPnoisy	1.2e-01	2.3e-02	4.0e-02	3.8e-02	3.8e-02	2.7e-02	2.6e-02	7.6e-02	**2.1e-02**	**2.1e-02**	**2.1e-02**
GMMcon	1.1e+03	1.1e+03	1.1e+03	1.1e+03	1.1e+03	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	1.1e+03	1.1e+03
GMMineq	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**	**1.0e+03**
trainANN	**7.2e+00**	7.4e+00	**7.2e+00**	**7.2e+00**	**7.2e+00**	**7.2e+00**	**7.2e+00**	**7.2e+00**	7.5e+00	7.7e+00	7.5e+00

[a]We multiplied the best solution value found with objective (MSE) with $|\mathcal{D}|$ to enable a comparison to the methods using objective (SSE).

D.3. Model equations

Ready-to-use implementations and datasets of the problems used for the benchmark study in Sections 3.6.2 and 3.6.3 are published in our open-source git repository GloPSE, cf. Section 3.1.3, while the mathematical formulations and the optimal solutions found are described in the following. For all models, we apply a reduced-space formulation [32, 140] by substituting all explicit equations into the objective function. With this, we obtain a solely box-constrained optimization problem for models EOS, IHMunc, kinetics, TSP, and trainANN. Note that we use an a-priori given start solution for the initialization of the B&B algorithm only for model trainANN.
In Sections 3.6.2 and 3.6.3, we discuss the final lower and upper bounds as well as the CPU times based on median values of 5 repetitive runs. Without loss of generality, we use the results of the first of the 5 runs to investigate the parameter values of the best solution found in this section.

D.3.1. "EIS": Electrochemical impedance spectroscopy

D.3.1.1. Problem formulation

In electrochemical impedance spectroscopy, the output signal of an electrochemical system to sinusoidal input signals is measured for different frequencies f [e.g., 121]. Subsequently, the input-output relation is modeled via a transfer function, e.g., the electrical impedance Z. As suggested by J. Raphael Seidenberg[1], we model a specific electrochemical system via a series circuit of resistor R_0 and two subcircuits $i = 1, 2$, each containing a constant phase element in parallel to a resistor R_i. This gives parameter estimation problem

$$
\begin{aligned}
\min_{\substack{R_0,R_1,R_2,\\ Q_1,Q_2,\alpha_1,\alpha_2}} \; & \sum_{(f^{\text{meas}},Z^{\text{meas}})\in\mathcal{D}} \left(\frac{||Z(f^{\text{meas}}) - Z^{\text{meas}}||}{0.01 \cdot ||Z^{\text{meas}}||} \right)^2 \\
\text{s.t.} \; & -\operatorname{Im}(Z(f)) = \sum_{i=1}^{2} \frac{\sin(\frac{\pi}{2}\cdot\alpha_i)\cdot R_i^2 \cdot x_i(f)}{\left(R_i \cdot x_i(f)\cdot\sin(\frac{\pi}{2}\cdot\alpha_i)\right)^2 + \left(R_i\cdot x_i(f)\cdot\cos(\frac{\pi}{2}\cdot\alpha_i)+1\right)^2} \\
& \operatorname{Re}(Z(f)) = R_0 + \sum_{i=1}^{2} \frac{\cos(\frac{\pi}{2}\cdot\alpha_i)\cdot R_i^2 \cdot x_i(f) + R_i}{\left(R_i \cdot x_i(f)\cdot\sin(\frac{\pi}{2}\cdot\alpha_i)\right)^2 + \left(R_i\cdot x_i(f)\cdot\cos(\frac{\pi}{2}\cdot\alpha_i)+1\right)^2} \\
& x_i(f) = Q_i \cdot \exp(\alpha_i \cdot \ln(2\cdot\pi\cdot f)) \qquad \forall i = 1,2 \\
& Q_1 \cdot R_1 \le Q_2 \cdot R_2
\end{aligned}
$$

with parameter bounds $R_0 \in [0.1, 0.15]$, R_1, $R_2 \in [0.001, 0.03]$, $Q_1 \in [0.001, 0.05]$, $Q_2 \in [0.1, 10]$, and α_1, $\alpha_2 \in [0.4, 0.999999]$. The inequality constraint is included for symmetry breaking. Note that we use the Euclidean norm $||\cdot||$ to measure the prediction error since the electrical impedance Z is a complex number with real part $\operatorname{Re}(Z)$ and imaginary part $\operatorname{Im}(Z)$. Note further that the *negative* imaginary part $-\operatorname{Im}(Z)$ is measured and modeled. Dataset $\mathcal{D}$ contains electrochemical data on alkaline water electrolysis obtained in a three-electrode beaker cell mimicking industrially relevant conditions, e.g., 80 °C and 30 wt. % KOH. Please refer to [198] for more details on the experimental setup. We use $|\mathcal{D}| = 26$

[1] Affiliation: Process Systems Engineering, RWTH Aachen University, 52074 Aachen, Germany

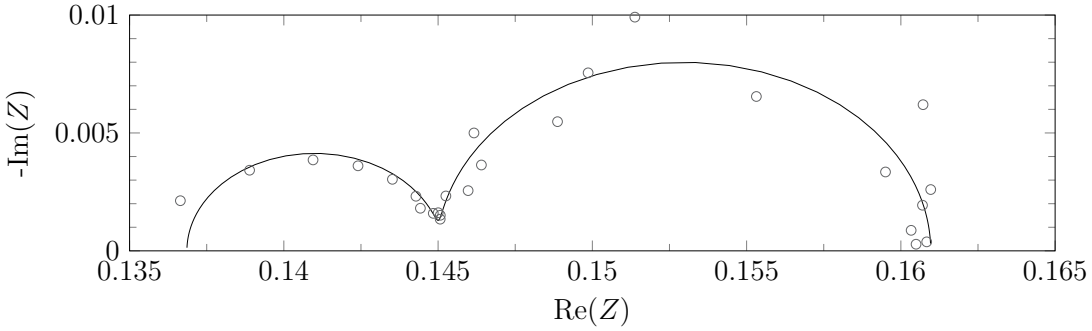


Figure D.1.: Best solution found for problem EIS (–) compared to the measurement data (o)

data points of the galvanostatic electrochemical impedance spectroscopy data obtained at $2\,\mathrm{A/m^2}$ by Thissen et al. (2024) [199].

D.3.1.2. Optimal solution

All approaches report solution point $R_0 = 0.13686$, $R_1 = 0.0081671$, $R_2 = 0.015939$, $Q_1 = 0.012061$, $Q_2 = 1.92067$, $\alpha_1 = 0.9999986$, and $\alpha_2 = 0.9999989$ which has been found in the root node. This solution is guaranteed to be a global optimum, since the standard B&B algorithm converges. The trajectory of the optimized estimation problem is shown in Figure D.1.

D.3.2. "EOS": Equation of state of propane

D.3.2.1. Problem formulation

In parameter estimations problems EOS262, EOS2262, EOS262noisy, and EOS2262noisy, we fit synthetic measurement data to an empirical equation of state of propane based on the Helmholtz energy proposed by Lemmon et al. [124]. In fact, we estimate 9 out of 162 unknown parameters considered in [124] by minimizing the prediction error on pressure values yielding

$$\begin{aligned}
\min_{\substack{n_{10},t_{10},d_{10},c_{10},l_{10},\\ \eta_{10},\beta_{10},\gamma_{10},\varepsilon_{10}}} \quad & \sum_{(p^{\mathrm{meas}},\tau^{\mathrm{meas}},\delta^{\mathrm{meas}})\in\mathcal{D}} \left(\frac{p^{\mathrm{meas}} - p(\delta=\delta^{\mathrm{meas}},\tau=\tau^{\mathrm{meas}})}{p^{\mathrm{meas}}}\right)^2 \\
\text{s.t.} \quad & p(\delta,\tau) = \rho_{\mathrm{c}}\delta\cdot R\cdot\frac{T_{\mathrm{c}}}{\tau}\cdot(1+\Lambda^{\mathrm{r}}_{01}) \\
& \Lambda^{\mathrm{r}}_{01} = \sum_{i=1}^{18} n_i\tau^{t_i}\delta^{d_i}\cdot(d_i - c_i l_i\delta^{l_i} - 2\eta_i\delta^2 + 2\eta_i\varepsilon_i\delta) \\
& \qquad \cdot\exp\left(-c_i\delta^{l_i} - \eta_i(\delta-\varepsilon_i)^2 - \beta_i(\tau-\gamma_i)^2\right)
\end{aligned}$$

with constants $\rho_{\mathrm{c}} = 5\,000\,\mathrm{mol/m^3}$, $T_{\mathrm{c}} = 369.89\,\mathrm{K}$, $R = 8.314\,472\,\mathrm{J/(mol\,K)}$, and parameters n_i, t_i, d_i, c_i, l_i, η_i, β_i, γ_i, ε_i with indices $i \in \{1,\ldots,9,11,\ldots,18\}$ fixed to the values given in Table 4 of [124], as well as parameter bounds $n_{10} \in [-3,2]$, $t_{10} \in [10^{-5},22]$, $d_{10} \in \{1,\ldots,10\}$, $c_{10} \in \{0,1\}$, $l_{10} \in \{1,2\}$, $\eta_{10} \in [0,20]$, $\beta_{10} \in [0,550]$, $\gamma_{10} \in [0,2]$, and $\varepsilon_{10} \in [0,1.5]$. The parameter bounds are based on parameter values for different fluids

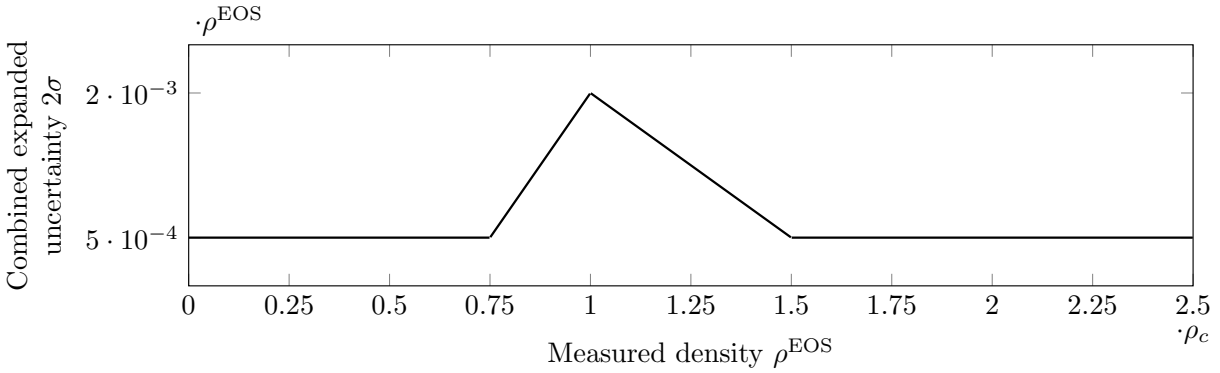


Figure D.2.: Assumed uncertainties 2σ for different densities in dependence of the distance to the critical density ρ_c

reported in [124, 184, 212]. Note that we allow for a Gaussian term in term number 10 via $c_{10} \in \{0, 1\}$. Further, we use the value reported in [124] for gas constant R rather than the standard value of the international system of units (SI) to stay consistent with the model including the parameter values given in [124]. The choice to optimize the parameters of addend $i = 10$ is a compromise between the excessively long runtimes for optimizing the shape of the first addends, which determine the general properties of the model predictions, and the almost instantaneous convergence when optimizing the shape of the last addends, which yield a fine tuning.

For EOS2262, we generated 2262 exact data points with the model based on the optimal parameter values of [124] representing the measurements. Each measurement is a triplet of pressure p, scaled density δ and scaled temperature τ. Out of these synthetic data points, we picked 262 data points evenly distributed over the pressure, density, and temperature domain for problem EOS262.

For problems EOS262noisy and EOS2262noisy, we generated pseudo-experimental data. For this, we consider the combined expanded uncertainties covering the uncertainties on the measurement of density, temperature, and pressure, and assume the absence of systematic errors. The actual uncertainty of experimental data varies between experimental datasets and research groups, cf. [124]. The best measurements available have a combined expanded uncertainty of about 0.05% for densities far away from the critical point. Approaching the critical point, the uncertainties increase, compare [20]. For simulating realistic measurement noise, we therefore use independent normally distributed errors $\varepsilon(\rho^{\mathrm{EOS}}) \sim \mathrm{N}(0, \sigma^2(\rho^{\mathrm{EOS}}))$ with standard deviation

$$\sigma(\rho^{\mathrm{EOS}}) = \frac{1}{2} \cdot \begin{cases} 0.0005 \cdot \rho^{\mathrm{EOS}} & \text{if } \rho^{\mathrm{EOS}} \notin [0.75 \cdot \rho_c, 1.5 \cdot \rho_c] \\ \left(0.002 + 0.006 \cdot (\frac{\rho^{\mathrm{EOS}}}{\rho_c} - 1)\right) \cdot \rho^{\mathrm{EOS}} & \text{if } \rho^{\mathrm{EOS}} \in [0.75 \cdot \rho_c, \rho_c] \\ \left(0.002 - 0.003 \cdot (\frac{\rho^{\mathrm{EOS}}}{\rho_c} - 1)\right) \cdot \rho^{\mathrm{EOS}} & \text{if } \rho^{\mathrm{EOS}} \in (\rho_c, 1.5 \cdot \rho_c] \end{cases} ,$$

see Figure D.2. The resulting error is added only to the density values yielding data

$$\begin{aligned} p^{\mathrm{exp}} &:= p^{\mathrm{EOS}} \\ T^{\mathrm{exp}} &:= T^{\mathrm{EOS}} \\ \rho^{\mathrm{exp}} &:= \rho^{\mathrm{EOS}} + \varepsilon(\rho^{\mathrm{EOS}}) \quad , \end{aligned}$$

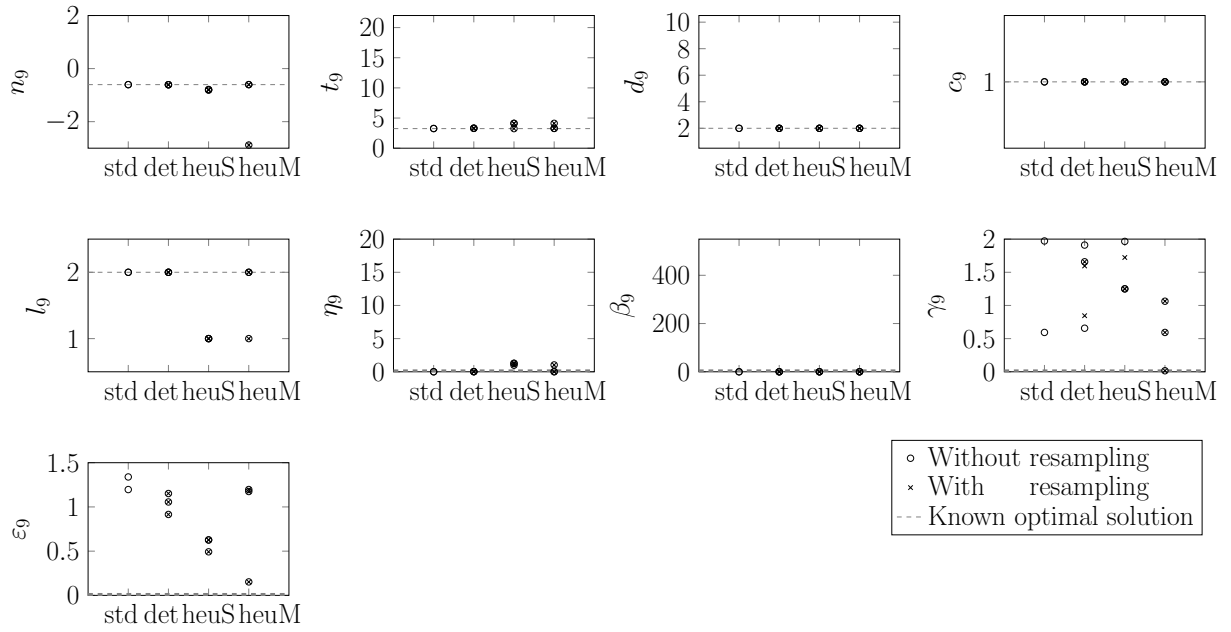


Figure D.3.: Best solution point found within a CPU time limit of 23 h for problem EOS262 with the standard B&B algorithm (std) as well as the deterministic approach (det), the SSE heuristic (heuS) and the MSE heuristic (heuM) of the B&B algorithm with growing datasets applying different augmentation rules compared to the optimal solution point used for generating the synthetic measurement data [124]

where $(p^{\mathrm{EOS}}, T^{\mathrm{EOS}}, \rho^{\mathrm{EOS}})$ is the exact synthetic measurement data used for problems EOS262 and EOS2262.

D.3.2.2. Optimal solution

The results of the EOS model show similar tendencies no matter whether using 262 or 2262 exact/noisy data points. As an example, we depict the optimal parameter values calculated for EOS262 and EOS2262noisy in Figures D.3 and D.4, respectively. When using exact measurement data, the standard B&B algorithm, the deterministic approach with any augmentation rule as well as the heuristic approaches for some of the augmentation rules converge. Consequently, we observe similar parameter values for these approaches in Figure D.3. Only exception are parameters γ_{10} and ε_{10} which have only marginal impacts on the objective function in comparison to the other unknown parameters.
When using noisy measurement data, all approaches hit the CPU time limit. Although the objective value of the best solution values found within the CPU time limit is similar for all approaches, the corresponding solution points differ significantly, see Figure D.4.

D.3.3. "GMM": Gaussian mixture models

D.3.3.1. Problem formulation

In Gaussian mixture models, we assume that the data points are realizations of K different clusters, where the data points of cluster k follow a Gaussian distribution with mean $\boldsymbol{\mu}_k$ and Σ_k for any $k = 1, \ldots, K$. The prior probability of a data point being part of

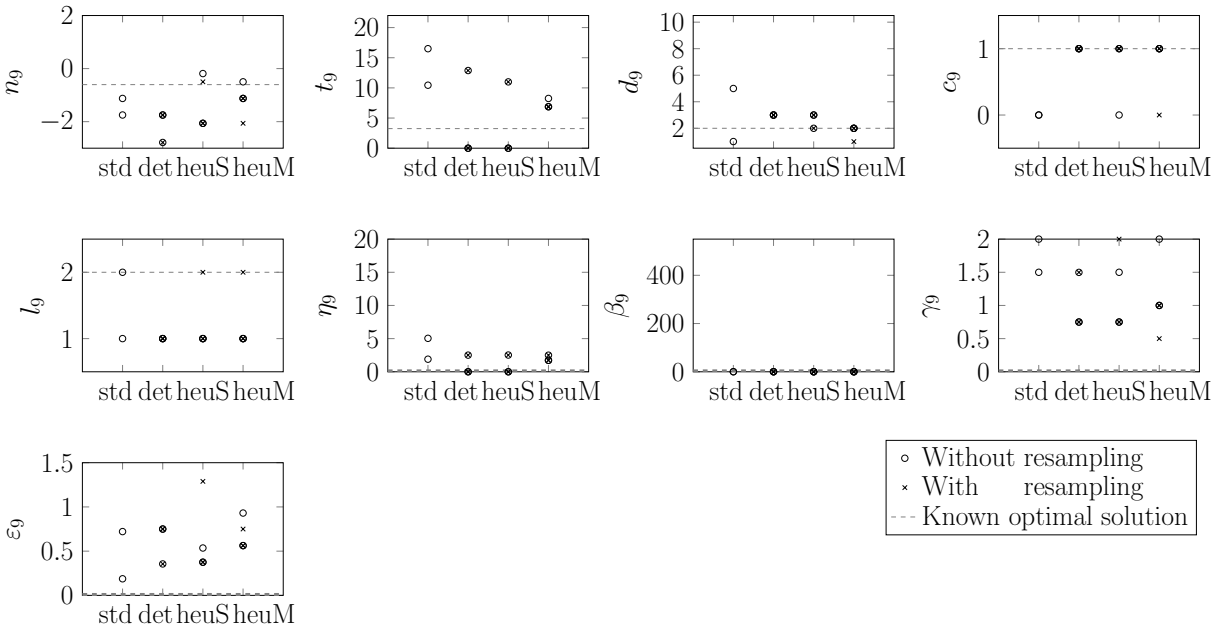


Figure D.4.: Best solution point found within a CPU time limit of 23 h for problem EOS2262noisy with the standard B&B algorithm (std) as well as the deterministic approach (det), the SSE heuristic (heuS) and the MSE heuristic (heuM) of the B&B algorithm with growing datasets applying different augmentation rules compared to the optimal solution point used for generating the synthetic measurement data [124]

cluster k is given by mixing coefficient ϕ_k, $k = 1, \ldots, K$. Commonly, the mixture coefficients as well as the unknown mean and covariance of the clusters are determined by the expectation-maximization (EM) algorithm [49] which is an iterative procedure for solving the maximum likelihood problem. We, in turn, solve the maximum likelihood problem with the B&B algorithm.
For simplicity, we restrict ourselves to 1-dimensional data points and $K = 2$ clusters, yielding maximum likelihood problem GMMcon

$$\min_{\substack{\phi_k,\mu_k,\sigma_k,\\ k=1,2}} \sum_{d=1}^{|\mathcal{D}|} Q_d$$

$$\begin{aligned}
\text{s.t.} \quad Q_d &= \sum_{k=1}^{2} \gamma_{d,k} \cdot l_{d,k} && \forall d = 1, \ldots, |\mathcal{D}| \\
\gamma_{d,k} &= \frac{\phi_k \cdot N_{d,k}}{\sum_{j=1}^{2} \phi_j \cdot N_{d,j}} && \forall d = 1, \ldots, |\mathcal{D}|,\ k = 1, 2 \\
l_{d,k} &= -\ln(\phi_k) - \ln(N_{d,k}) && \forall d = 1, \ldots, |\mathcal{D}|,\ k = 1, 2 \\
N_{d,k} &= \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_k^2}} e^{-\frac{1}{2}\left(\frac{x_d - \mu_k}{\sigma_k}\right)^2} && \forall d = 1, \ldots, |\mathcal{D}|,\ k = 1, 2 \\
1 &= \sum_{k=1}^{2} \phi_k \\
\mu_2 &\leq \mu_1
\end{aligned}$$

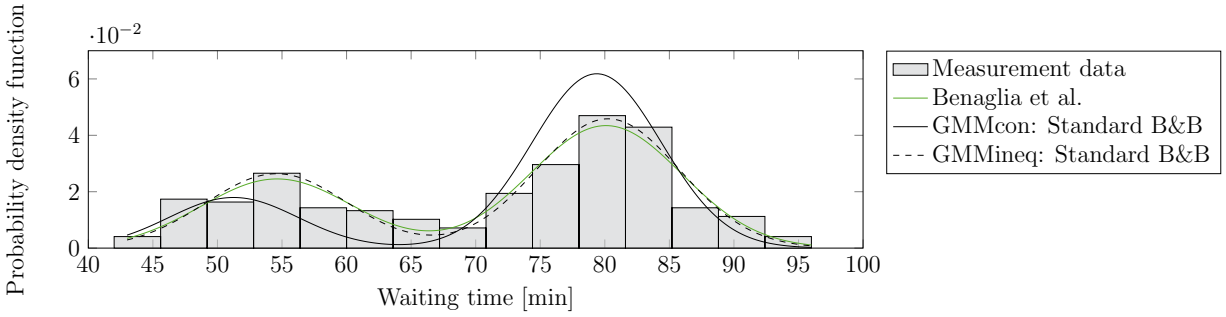


Figure D.5.: Cumulative probability function based on the solutions of Benaglia et al. [23] and of the standard B&B algorithm compared to a histogram of the measurement data

with data points $\mathcal{D} = \{x_1, x_2, \ldots\} \subsetneq \mathbb{R}$. The inequality constraint on the means is introduced for breaking the symmetry. Note that l_{d_k} is the negative log-likelihood function such that we solve a minimization problem.
Equivalently, we use the equality constraint on the mixing coefficients for eliminating one of the unknown parameters. We obtain model GMMineq

$$
\begin{aligned}
&\min_{\substack{\phi_1,\mu_k,\sigma_k,\\ k=1,2}} \sum_{d=1}^{|\mathcal{D}|} Q_d \\
\text{s.t.} \quad Q_d &= \sum_{k=1}^{2} \gamma_{d,k} \cdot l_{d,k} && \forall d = 1, \ldots, |\mathcal{D}| \\
\gamma_{d,k} &= \frac{\phi_k \cdot N_{d,k}}{\sum_{j=1}^{2} \phi_j \cdot N_{d,j}} && \forall d = 1, \ldots, |\mathcal{D}|,\ k = 1,2 \\
l_{d,k} &= -\ln(\phi_k) - \ln(N_{d,k}) && \forall d = 1, \ldots, |\mathcal{D}|,\ k = 1,2 \\
N_{d,k} &= \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_k^2}} e^{-\frac{1}{2}\left(\frac{x_d - \mu_k}{\sigma_k}\right)^2} && \forall d = 1, \ldots, |\mathcal{D}|,\ k = 1,2 \\
\phi_2 &= 1 - \phi_1 \\
\mu_2 &\leq \mu_1
\end{aligned}
$$

As the measurement data for both GMMcon and GMMineq, we use the measured waiting time between two eruptions of geyser Old faithful in Yellowstone National Park, Wyoming, USA [13, 88] containing 272 data points in total. Based on these data points, we use parameter bounds $\sigma_k \in [5, 300]$ as well as $\mu_k \in [40, 100]$ for $k = 1, 2$. We bound the unknown mixture coefficient by $\phi_k \in [0.1, 0.9]$, $k = 1, 2$ or $k = 1$ to ensure positive arguments of $\ln(\cdot)$ even in presence of numerical tolerances.

D.3.3.2. Optimal solution

For problem GMMcon, the standard B&B algorithm gives the same solution after 23 h independent of the choice between objective (SSE) and (MSE), while there are minor deviations in the solutions of the different approaches for the B&B algorithm with growing datasets, see Table D.4. Note that there are approaches of the B&B algorithm with growing datasets obtaining both tighter, i.e., larger, lower bounds and tighter, i.e., smaller, upper

Table D.4.: Best solution points found for problems GMMcon and GMMineq within a CPU time limit of 23 h by the standard B&B algorithm and the range of the optimal values for the unknown parameters obtained by any approach of the B&B algorithm with growing datasets compared to the values reported in the literature. The values of mean and standard deviation are waiting times in minutes rounded to one decimal place.

	ϕ_1	μ_1	σ_1	ϕ_2	μ_2	σ_2
Benaglia et al. [23]	0.63915	80.1	5.9	0.36085	54.6	5.9
GMMcon						
Standard B&B	0.775	79.4	5.0	0.225	51.3	5.0
B&B with growing datasets	[0.475, 0.75]	[79.4, 81.3]	[5.0, 6.0]	[0.25, 0.525]	[53.7, 55.0]	[5.0, 6.5]
GMMineq						
Standard B&B (SSE)	0.6362	80.2	5.5	–	54.6	5.5
Standard B&B (MSE)	0.6359	80.3	5.5	–	54.6	5.6
B&B with growing datasets	[0.60, 0.65]	[79.4, 81.0]	[5.3, 6.2]	–	[54.2, 55.0]	[5.5, 6.1]

bounds, in particular the SSE heuristic. The solutions obtained for problem GMMineq by the all of the different approaches are even closer to each other. In fact, all parameter values computed are similar to the values reported in the literature [e.g., 23]. In particular, the best solution we could obtain for fitting a Gaussian mixture model to this dataset, namely the solution of the standard B&B algorithm with objective (SSE) for problem GMMineq, almost coincides with the solution from Benaglia et al. [23]. Figure D.5 shows the effect of the differences between the solution points on the probability function cumulated over the two Gaussian distributions. For comparison, we included the solution of the standard B&B algorithm obtained with problem GMMcon yielding a significantly higher, i.e., worse, objective value.

D.3.4. "IHM": Indirect Hard Modeling

D.3.4.1. Problem formulation

In models IHMcon and IHMunc, we fit spectral data for mixtures of multiple chemical species, namely components $k \in \mathcal{K}$, measured in dependence of wavenumber ν by Raman spectroscopy. Following the approach of Indirect Hard Modeling [8], we model Raman intensity $\mathcal{A}$ of the spectral mixture hard model (HM) as a weighted sum of spectral pure component hard models (PCM) $\mathcal{A}_k$ and a linear baseline model

$$\mathcal{A} = \sum_{k \in \mathcal{K}} w_k \cdot \mathcal{A}_k(\nu) \; + \; \Theta_{B,1} \; + \; \nu \cdot \Theta_{B,2} \; ,$$

where the flexible spectral PCM of each component k is given by a sum of Pseudo-Voigt functions over peaks $i \in \mathcal{I}(k)$

$$\mathcal{A}_k(\nu) = \sum_{i\in\mathcal{I}(k)} V_{k,i}(\nu) = \sum_{i\in\mathcal{I}(k)} \beta_{k,i}\cdot\alpha_{k,i}\cdot\exp\big(\frac{-\ln(2)\cdot(\nu-\delta_{k,i})^2)}{\gamma_{k,i}^2}\big) + (1-\beta_{k,i})\cdot\alpha_{k,i}\frac{\gamma_{k,i}^2}{(\nu-\delta_{k,i})^2+\gamma_{k,i}^2}$$

with peak maximum $\alpha_{k,i}$ at the peak's center position $\delta_{k,i}$, Gaussian-Lorentzian-ratio $\beta_{k,i}$, and half width at half maximum (HWHM) $\gamma_{k,i}$.
The resulting optimization model is challenging even for local solvers as it contains a large number of unknown parameters, cf. [116]. Thus, we reduce the problem size by

- choosing a measurement series containing only 2 components
- manually reducing the number of Pseudo-Voigt functions to be fitted,
- fixing some of the unknown parameters to the optimal solution point of [56] calculated with PEAXACT [168], namely the peak's center position $\delta_{k,i}$ and Gaussian-Lorentzian-ratio $\beta_{k,i}$ for any $k \in \mathcal{K}$, $i \in \mathcal{I}(k)$ as well as baseline parameters $\Theta_{B,1}$ and $\Theta_{B,2}$,
- merging weight w_k with $\alpha_{k,i}$ for any $k \in \mathcal{K}$.

Since the area under the peak curve is proportional to the concentration of the component [25, 135], and the concentrations should not change during parameter estimation, we demand a constant area

$$\alpha_{k,i}\cdot\gamma_{k,i}\cdot\Big(\beta_{k,i}\cdot\left(\frac{\ln(2)}{\pi}\right)^{\frac{-1}{2}} + (1-\beta_{k,i})\cdot\pi\Big) = \text{constant} \quad \forall k\in\mathcal{K},\, i\in\mathcal{I}(k)\ ,$$

compare also the PEAXACT manual [168].
Altogether, model IHMcon is given by estimation problem

$$\begin{aligned}
\min_{\substack{\tilde\alpha_{k,i},\gamma_{k,i}\\ \text{for } k\in\mathcal{K},\, i\in\mathcal{I}(k)}} \quad & \sum_{(\nu^{\text{meas}},\mathcal{A}^{\text{meas}})\in\mathcal{D}} \left(\mathcal{A}^{\text{meas}} - \big(\sum_{k\in\mathcal{K}}\mathcal{A}_k(\nu=\nu^{\text{meas}}) + \Theta_{B,1}^{\text{ref}} + \nu|_{\nu=\nu^{\text{meas}}}\cdot\ \Theta_{B,2}^{\text{ref}}\big)\right)^2 \\
\text{s.t.} \quad & \mathcal{A}_k(\nu) = \sum_{i\in\mathcal{I}(k)} V_{k,i}(\nu) && \forall k\in\mathcal{K} \\
& V_{k,i}(\nu) = \beta_{k,i}^{\text{ref}}\cdot\tilde\alpha_{k,i}\cdot\exp\big(\frac{-\ln(2)\cdot(\nu-\delta_{k,i}^{\text{ref}})^2)}{\gamma_{k,i}^2}\big) \\
& \qquad + (1-\beta_{k,i}^{\text{ref}})\cdot\tilde\alpha_{k,i}\cdot\frac{\gamma_{k,i}^2}{(\nu-\delta_{k,i}^{\text{ref}})^2+\gamma_{k,i}^2} && \forall k\in\mathcal{K},\ i\in\mathcal{I}(k) \\
& \tilde\alpha_{k,i}\cdot\gamma_{k,i} = w_k^{\text{ref}}\cdot\alpha_{k,i}^{\text{ref}}\cdot\gamma_{k,i}^{\text{ref}} && \forall k\in\mathcal{K},\, i\in\mathcal{I}(k)\ ,
\end{aligned}$$

with parameter bounds $\tilde\alpha_{k,i} \in [0, 10^4]$ and $\gamma_{k,i} \in [1, 100]$ for any $k \in \mathcal{K} = \{1, 2\},\ i \in \mathcal{I}(k)$ with $\mathcal{I}(1) = \{1, 2, 3\}$, and $\mathcal{I}(2) = \{1, \ldots, 7\}$, as well as superscript "ref" indicating the reference solution taken from [56].

Equivalently, we use the equality constraint to fix parameters $\gamma_{k,i}$ based on the determined values of parameters $\tilde{\alpha}_{k,i}$ for any $k \in \mathcal{K}$, $i \in \mathcal{I}(k)$. We obtain model IHMunc, an unconstrained optimization problem with less degrees of freedom

$$
\begin{aligned}
\min_{\substack{\tilde{\alpha}_{k,i} \\ \text{for } k\in\mathcal{K},\, i\in\mathcal{I}(k)}} \quad & \sum_{(\nu^{\text{meas}},\mathcal{A}^{\text{meas}})\in\mathcal{D}} \left(\mathcal{A}^{\text{meas}} - \big(\sum_{k\in\mathcal{K}} \mathcal{A}_k(\nu = \nu^{\text{meas}}) \;+\; \Theta^{\text{ref}}_{B,1} \;+\; \nu|_{\nu=\nu^{\text{meas}}} \cdot \Theta^{\text{ref}}_{B,2} \big) \right)^2 \\
\text{s.t.} \quad & \mathcal{A}_k(\nu) = \sum_{i\in\mathcal{I}(k)} V_{k,i}(\nu) && \forall k \in \mathcal{K} \\
& V_{k,i}(\nu) = \beta^{\text{ref}}_{k,i} \cdot \tilde{\alpha}_{k,i} \cdot \exp\big(\frac{-\ln(2) \cdot (\nu - \delta^{\text{ref}}_{k,i})^2)}{\gamma(\tilde{\alpha}_{k,i})^2} \big) \\
& \qquad + (1 - \beta^{\text{ref}}_{k,i}) \cdot \tilde{\alpha}_{k,i} \cdot \frac{\gamma(\tilde{\alpha}_{k,i})^2}{(\nu - \delta^{\text{ref}}_{k,i})^2 + \gamma(\tilde{\alpha}_{k,i})^2} && \forall k \in \mathcal{K},\; i \in \mathcal{I}(k) \\
& \gamma(\tilde{\alpha}_{k,i}) = \frac{w^{\text{ref}}_k \cdot \alpha^{\text{ref}}_{k,i} \cdot \gamma^{\text{ref}}_{k,i}}{\tilde{\alpha}_{k,i}} && \forall k \in \mathcal{K},\, i \in \mathcal{I}(k)
\end{aligned}
$$

with parameter bounds $\tilde{\alpha}_{k,i} \in [0, 10^4]$ for any $k \in \mathcal{K} = \{1, 2\}$, $i \in \mathcal{I}(k)$ with $\mathcal{I}(1) = \{1, 2, 3\}$, and $\mathcal{I}(2) = \{1, \ldots, 7\}$.
For both IHMcon and IHMunc, we use Raman spectra of formic acid in aqueous solution which were experimentally acquired for different titration steps [56, 57]. We choose the first titration step such that we have only two components. Additionally, we reduce the size of the dataset by picking every 10^{th} data point of the Raman spectrum yielding $|\mathcal{D}| = 284$ data points.

D.3.4.2. Optimal solution

For both IHMcon and IHMunc, all approaches lead to the same solution, namely the solution found in the root node, see Table D.5. Due to our restriction to fit only 10 instead of 16 Pseudo-Voigt functions, we cannot obtain the accuracy reported by Echtermeyer et al. [56]. When restricting the mixture HM calculated by [56] to the baseline and the 10 Pseudo-Voigt functions we have optimized, we obtain a reference solution which differs significantly from the measurement data in range $\nu \in [1\,000, 1\,350]$ and $\nu \in [1\,450, 1\,550]$, see Figure D.6. The reference solution differs to the mixture HM given by the solutions of IHMcon and IHMunc mainly around $\nu = 1350$ and in range $\nu \in [1\,450, 1\,550]$. The deviations around $\nu = 1350$ are caused by differences for 3 overlapping peaks at positions $\delta_{k,i} \in [1350, 1391]$, namely peak $i = 3$ of component $k = 1$ as well as peaks $i \in \{4, 5\}$ of component $k = 2$. The deviations in range $\nu \in [1\,450, 1\,550]$ are caused by differences for peak $i = 2$ of component $k = 2$ at position $\delta_{2,2} = 1510$, where the mixture HM's of IHMcon and IHMunc are more consistent to the measurement data. As a result, we obtain mixture HM's which are slightly closer to the measurements than the reference solution, where IHMcon yields a better fit to peak $i = 5$ of component $k = 2$ at position $\delta_{2,5} = 1350$. In return, the mixture HM given by IHMunc is a bit closer to the measurements just before this peak, i.e., in range $\nu \in [1300, 1340]$. With this, the best solutions of IHMcon and IHMunc achieve a root mean squared error which equals 98% and 97%, respectively, of the root mean squared error of the reference solution.

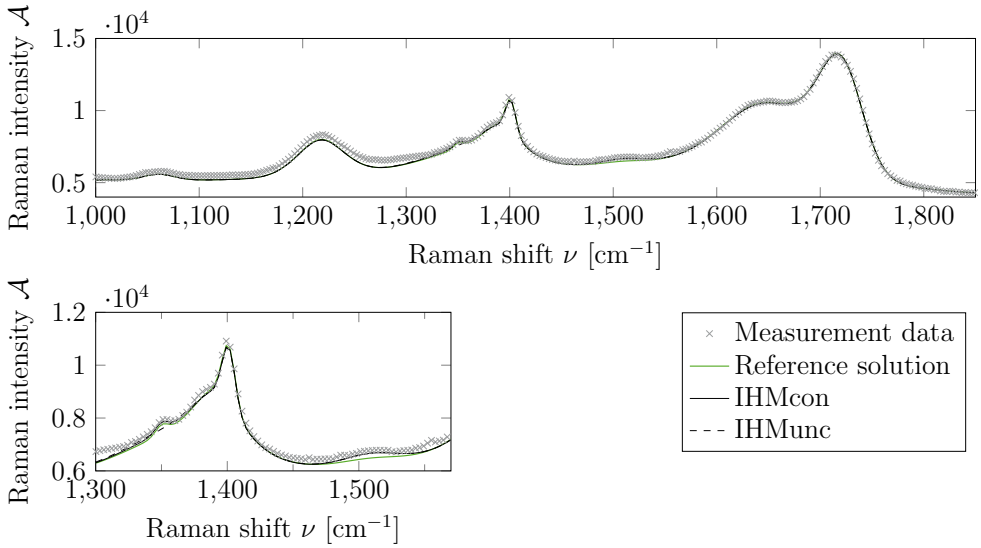


Figure D.6.: Spectral mixture HM given by best solution found within a CPU time limit of 23 h for problems IHMcon and IHMunc over spectral range $\nu \in [1\,000, 1\,850]$ and $\nu \in [1\,300, 1\,570]$ compared to the measurement data and a reference solution which is given by [56] reduced to the baseline and the Pseudo-Voigt functions of the 10 peaks $\mathcal{I}(1) = \{1, 2, 3\}$ and $\mathcal{I}(2) = \{1, \ldots, 7\}$

Table D.5.: Parameter values of the best solution found for IHMcon and IHMunc within a CPU time limit of 23 h and reference values of [56] for the 10 Pseudo-Voigt functions optimized

Problem	k	$i \in \mathcal{I}(k)$						
Peak positions $\boldsymbol{\delta_{k,i}}$								
Reference [56]	1	1643	1510	1354				
	2	1722	1704	1401	1391	1350	1217	1061
Gaussian-Lorentzian-ratio $\boldsymbol{\beta_{k,i}}$								
Reference [56]	1	0.2925	0.0001	0.0001				
	2	0.9999	0.9999	0.9999	0.0001	0.9999	0.7027	0.9999
Modified peak maximum $\tilde{\boldsymbol{\alpha}}_{\boldsymbol{k,i}}$								
Reference [56]	1	5904	920	1818				
	2	5525	2476	2167	2665	337	2654	408
IHMcon	1	5882	1106	1908				
	2	5531	2459	2175	2566	329	2628	410
IHMunc	1	5885	1107	1967				
	2	5531	2460	2180	2509	15	2631	410
HWHM $\boldsymbol{\gamma_{k,i}}$								
Reference [56]	1	58.81	53.20	75.02				
	2	22.84	31.02	5.79	25.59	5.67	27.39	17.64
IHMcon	1	59.04	44.28	71.47				
	2	22.81	31.24	5.77	26.58	5.81	27.66	17.55
IHMunc[a]	1	59.00	44.22	69.34				
	2	22.81	31.22	5.76	27.19	126.37	27.63	17.54

[a]For completeness, we show the values of $\gamma_{k,i}$ calculated based on the reported values of $\tilde{\alpha}_{k,i}$.

D.3.5. "kinetics": Kinetics of a chemical reaction

D.3.5.1. Problem formulation

We revisit our model "kinetic mechanism" of [140] for fitting the kinetic parameters of a chemical reaction of cyclohexadienyl radicals C_6H_7 discussed by Singer et al. [188] and Taylor et al. [197]. As in [140], we use the explicit Euler scheme to transform the dynamic optimization problem into an algebraic NLP. Apart from this, we extend the optimization problem by optimizing all 5 unknown parameters and obtain

$$
\begin{aligned}
\max_{k_i,\, i=1,\dots,5} \quad & \sum_{(t^{\text{meas}},y^{\text{meas}})\in\mathcal{D}} (y(t^{\text{meas}}) - y^{\text{meas}})^2 \\
\text{s.t.} \quad & \frac{x_A(t_{j+1}) - x_A(t_j)}{t_{j+1} - t_j} = k_1 \cdot x_Z(t_j) \cdot x_Y(t_j) - x_{O_2} \cdot (k_2 + k_3) \cdot x_A(t_j) \\
& \qquad + \frac{k_2}{K_2} \cdot x_D(t_j) + \frac{k_3}{K_3} \cdot x_B(t_j) - k_5 \cdot (x_A(t_j))^2 \quad \forall j = 0, \dots, 445 \\
& \frac{x_B(t_{j+1}) - x_B(t_j)}{t_{j+1} - t_j} = k_3 \cdot x_{O_2} \cdot x_A(t_j) - (\frac{k_3}{K_3} + k_4) \cdot x_B(t_j) \quad \forall j = 0, \dots, 445 \\
& \frac{x_D(t_{j+1}) - x_D(t_j)}{t_{j+1} - t_j} = k_2 \cdot x_{O_2} \cdot x_A(t_j) - \frac{k_2}{K_2} \cdot x_D(t_j) \quad \forall j = 0, \dots, 445 \\
& \frac{x_Y(t_{j+1}) - x_Y(t_j)}{t_{j+1} - t_j} = -k_1 \cdot 10^{-6} \cdot x_Z(t_j) \cdot x_Y(t_j) \quad \forall j = 0, \dots, 445 \\
& \frac{x_Z(t_{j+1}) - x_Z(t_j)}{t_{j+1} - t_j} = -k_1 \cdot x_Z(t_j) \cdot x_Y(t_j) \quad \forall j = 0, \dots, 445 \\
& y(t_j) = x_A(t_j) + \frac{2}{21} x_B(t_j) + \frac{2}{21} x_D(t_j) \quad \forall j = 0, \dots, 446 \\
& x_A(t_0) = 0 \\
& x_B(t_0) = 0 \\
& x_D(t_0) = 0 \\
& x_Y(t_0) = 0.4 \\
& x_Z(t_0) = 140
\end{aligned}
$$

with initial time $t_0 = 0$, final time $t_f = 4.46$ and constant step width $\Delta t := t_{j+1} - t_j = 0.01$, $j = 0, \dots, 445$, constants $T = 273$, $K_2 = 46 \cdot \exp(\frac{6500}{T} - 18)$, $K_3 = 2 \cdot K_2$, and $x_{O_2} = 0.002$, as well as parameter bounds $k_1, k_2, k_3 \in [10, 1200]$ and $k_4, k_5 \in [0.001, 40]$. We use the data of [140] which gives a measurement of output y for any discretized time point t_j, $j = 0, \dots, 446$. Note that we fix the initial value at time $t_0 = 0$ to the respective measurement rather than optimizing it. We therefore obtain dataset $\mathcal{D} = \{(0.01 \cdot j, y_j^{\text{meas}}) \in \mathbb{R}^2 : j = 1, \dots, 446\}$.

D.3.5.2. Optimal solution

Mitsos et al. [140] and Singer et al. [188] calculated the optimal solution point $k_2 = 828.0$, $k_3 = 394.0$, and $k_4 = 13.5$ with fixed parameters $k_1 = 53$ and $k_5 = 0.0012$. Note that Singer et al. [188] calculate the global solution of a dynamic optimization problem, whereas we and Mitsos et al. [140] calculate the global solution of the discretized model using the

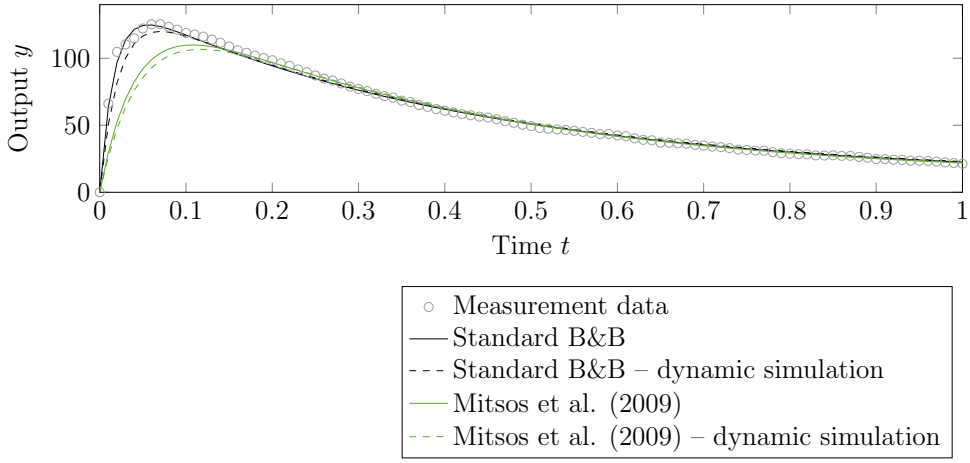


Figure D.7.: Output trajectory for the best solution found by the standard B&B algorithm and the optimal solution of Mitsos et al. (2009) [140] of the discretized model as well as the simulated trajectory using MATLAB for integrating the corresponding dynamic system compared to the measurement data used for optimization

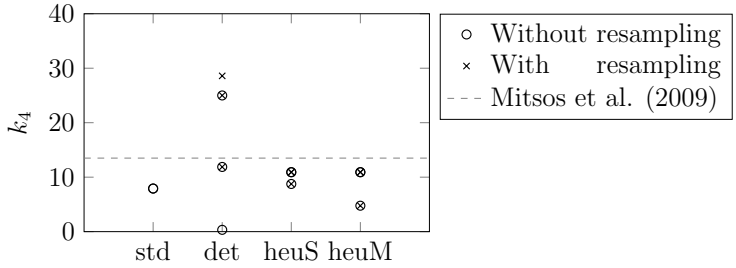


Figure D.8.: Values of parameter k_4 of the best solution found by the standard B&B algorithm as well as the deterministic approach, the SSE heuristic and the MSE heuristic of the B&B algorithm with growing datasets applying different augmentation rules compared to the parameter value of the optimal solution reported by [140]

explicit Euler scheme with constant step width $\Delta t = 0.01$. The effect of the numerical integration errors is illustrated in Figure D.7, where a more accurate dynamic simulation is calculated for comparison by applying the solver `ode45` of MATLAB® implementing an integration scheme of higher order.

In contrast to [140], we solve for all five reaction rates $k_1, \ldots, k_5$. Moreover, we do not reach convergence in our case study. Not surprisingly, our solution point obtained by the standard B&B algorithm, , namely $k_1 = 113.6345$, $k_2 = 783.0214$, $k_3 = 378.5887$, $k_4 = 7.9134$, and $k_5 = 0.0010$, differs from the solution of [140]. The additional degrees of freedom allow for a more accurate fit at the initial phase of the reaction $t \in [0, 0.2]$, see Figure D.7. Note that all trajectories overlap with the data points in the interval $t \in [1, 4.46]$ which are not depicted. Note further that we obtain the same solution with the standard B&B algorithm independent of the choice between objective (SSE) and (MSE).

The best objective value found with the deterministic approach applying augmentation rules SCALING and COMBI is larger, i.e., worse, than the best objective value found by the standard B&B algorithm by a factor of about 40 and 3, respectively, without resampling and about 40 and 2, respectively, with resampling. Similarly, we see significant deviations

Table D.6.: Best solution points found for problem trainANN within a CPU time limit of 23 h by the standard B&B algorithm (std) as well as the deterministic (det) and heuristic approaches for the B&B algorithm with growing datasets

Approaches[a]	k	j	$w_{j,1}^{(k)}$	$w_{j,2}^{(k)}$	$w_{j,3}^{(k)}$	$w_{j,4}^{(k)}$	$w_{j,5}^{(k)}$
std with (MSE), MSE heuristic with TOL	1	1	0	0	0	0.25	0
	1	2	0.233	0.233	0.482	0.187	0.094
	2	–	0	0	0.230	0.365	0.360
MSE heuristic with CONST	1	1	0.078	0.078	0.060	0.041	0.021
	1	2	0.237	0.487	0.177	0.118	0.059
	2	–	0	0.275	0.264	0.257	0.253
MSE heuristic with OOS	1	1	0.073	0.204	0.499	0.465	0.425
	1	2	0.793	0.420	0.533	0.437	0.387
	2	–	0.725	0.192	0.030	0.021	0.020
std with (SSE), det[b], SSE heuristic[b]	1	1	0.075	0.043	0.158	0.542	0.420
	1	2	0.602	0.993	0.474	0.485	0.796
	2	–	0.000	0.814	0.003	0.000	0.000

	$b_1^{(1)}$	$b_2^{(1)}$	$b_3^{(1)}$	$b_4^{(1)}$	$b_5^{(1)}$	$b^{(2)}$
std with (MSE), MSE heuristic with TOL	0	0	0	0	0	0.009
MSE heuristic with CONST	0	0	0	0	0	0
MSE heuristic with OOS	0.012	0.042	0.338	0.463	0.541	0.051
std with (SSE), det[b], SSE heuristic[b]	0.000	0.000	0.001	0.090	0.088	0.000

[a]Solutions are the same when only changing between with and without resampling.

[b]Solutions are the same for all augmentation rules.

in the corresponding parameter values. For all other approaches of the B&B algorithm with growing datasets, the parameter values of k_1, k_2, and k_3 deviate by at most 4% from the best values found by the standard B&B algorithm. The corresponding values of parameter k_5 range in $[0.001, 0.0014]$, where the solution reported by the standard B&B algorithm gives $k_5 = 0.001$ and the fixed value of [140, 188] is $k_5 = 0.0012$. Only for parameter k_4 we can see large deviations for the different approaches, see Figure D.8. However, in most cases, the B&B algorithm with growing datasets yields for k_4 values between the optimal value $k_4 = 7.91$ determined by the standard B&B algorithm and optimal value $k_4 = 13.5$ from the literature [140, 188].

D.3.6. "trainANN": Training an artificial neural network

D.3.6.1. Problem formulation

We train an artificial neural network (ANN) with an input layer containing 2 inputs, one hidden layer with 5 neurons, and an output layer covering 1 output. For the activation function, we choose the hyperbolic tangent. Note that MAiNGO implements envelopes for the hyperbolic tangent allowing for tight convex relaxations of the ANN build on top of these [181]. To improve the speed of convergence further, we apply weight decay as a

Table D.7.: Performance of the different ANN's calculated by the standard B&B algorithm (std) as well as the deterministic (det) and heuristic approaches for the B&B algorithm with growing datasets measured via the mean squared error of the training and validation set.

Approaches[a]	MSE	Maximum error	MSE	Maximum error
	Training set		Validation set	
std with (MSE), MSE heuristic with TOL	0.0692	0.2862	0.0626	0.3963
MSE heuristic with CONST	0.0739	0.2588	0.0676	0.3713
MSE heuristic with OOS	0.0479	0.5256	0.0421	0.6462
std with (SSE), det[b], SSE heuristic[b]	0.0409	0.4862	0.0353	0.5951

[a]Solutions are the same when only changing between with and without resampling.

[b]Solutions are the same for all augmentation rules

regularization [95], i.e., we add $\lambda \cdot ||\boldsymbol{w}||^2 = \sum_{d=1}^{|\mathcal{D}|} \frac{\lambda}{|\mathcal{D}|} \cdot ||\boldsymbol{w}||^2$ with regularization parameter λ and vector $\boldsymbol{w}$ containing all weights of the ANN to the objective function. We fix the regularization parameter to $\lambda = 0.01$ after manually investigating the changes of the lower bounds over all nodes calculated for the first few B&B iterations in dependence of different values for λ.
We obtain parameter estimation problem

$$\begin{aligned}
\min_{\substack{\boldsymbol{w},b^{(2)} \\ b_i^{(1)},i=1,\dots,5}} \quad & \sum_{(\boldsymbol{x}^{\text{meas}},y^{\text{meas}})\in\mathcal{D}} \left(\left(y(\boldsymbol{x}^{\text{meas}}) - y^{\text{meas}}\right)^2 + \frac{0.01}{|\mathcal{D}|} \cdot ||\boldsymbol{w}||^2 \right) \\
\text{s.t.} \quad v_i(\boldsymbol{x}) &= \sum_{j=1}^{2} w_{j,i}^{(1)} \cdot x_j + b_i^{(1)} && \forall i = 1,\dots,5 \\
z_i(\boldsymbol{x}) &= \tanh(v_i(\boldsymbol{x})) && \forall i = 1,\dots,5 \\
y(\boldsymbol{x}) &= \sum_{j=1}^{5} w_j^{(2)} \cdot z_i(\boldsymbol{x}) + b^{(2)}
\end{aligned}$$

with weight vector $\boldsymbol{w} = (w_{1,1}^{(1)},\dots,w_{1,5}^{(1)},w_{2,1}^{(1)},\dots,w_{2,5}^{(1)},w_1^{(2)},\dots,w_5^{(2)}) \in [0,1]^{15}$, biases $(b_1^{(1)},\dots,b_5^{(1)},b^{(2)}) \in [0,10]^6$, and normalized data $(\boldsymbol{x}^{\text{meas}},y^{\text{meas}}) \in [0,1]^3$ for any $(\boldsymbol{x}^{\text{meas}},y^{\text{meas}}) \in \mathcal{D}$. As a start solution, we use random values between 0 and 1 for the weights, and 0 for the biases.
We train the ANN to predict the total number of bikes rented based on the month and daily temperature given by real bike sharing data [62, 63]. We randomly pick about 30% of the data points for the training set yielding $|\mathcal{D}| = 220$. The data points in the training set are shuffled to improve the generalization performance. The remaining data points are saved in the validation set used for evaluating the results in the following subsection.

D.3.6.2. Optimal solution

When optimizing problem trainANN with the different approaches discussed in Chapter 3, we obtain 4 distinct solutions, see Table D.6. Note that the objective values corresponding to these solution points differ. Moreover, the solution point may differ from the listed values if minor variations in the CPU times for processing a node lead to a different number of

nodes processed within the fixed CPU time limit. However, both the mean squared error and the maximum error are significantly smaller than 1 for all solutions. Thus, we can predict the number of rental bikes well. Moreover, all ANN's generalize well. In fact, the mean squared error over the validation set is even smaller than over the training set. Contrarily, the maximum of the prediction error for each data point rises when replacing the training data with the validation data, but only by about 0.1. Note that we calculated the errors of Table D.7 with MATLAB®. The values may therefore slightly differ from values reported by MAiNGO due to numerical rounding errors.

D.3.7. "TSP": Metabolic pathways

D.3.7.1. Problem formulation

We revisit the model for fitting metabolic pathways from [142] which is included in the benchmark library BioPreDyn-bench [204] under the name "TSP". Since the TSP model is a dynamic optimization problem, we use the explicit Euler scheme to obtain the algebraic NLP

$$\min_{\substack{Ki_1,Ki_2,Ki_3,Ka_1,Ka_2,Ka_3,\\ K_4,K_5,K_6,V_1,V_2,V_3}} \quad \sum_{d=1}^{|\mathcal{D}|}\Bigg(\sum_{i=1}^{3}\left(E_{i,d}^{\text{meas}}-E_i(t_d)\right)^2+\sum_{i=1}^{3}\left(G_{i,d}^{\text{meas}}-G_i(t_d)\right)^2 +\sum_{i=1}^{2}\left(M_{i,d}^{\text{meas}}-M_i(t_d)\right)^2\Bigg)$$

$$\text{s.t.} \quad \frac{G_1(t_{j+1})-G_1(t_j)}{t_{j+1}-t_j}=\frac{V_1}{1+(\frac{P}{Ki_1})^2+(\frac{Ka_1}{S})^2}-G_1(t_j) \qquad \forall j=0,\dots,119$$

$$\frac{G_2(t_{j+1})-G_2(t_j)}{t_{j+1}-t_j}=\frac{V_2}{1+(\frac{P}{Ki_2})^2+(\frac{Ka_2}{M_1(t_j)})^2}-G_2(t_j) \qquad \forall j=0,\dots,119$$

$$\frac{G_3(t_{j+1})-G_3(t_j)}{t_{j+1}-t_j}=\frac{V_3}{1+(\frac{P}{Ki_3})^2+(\frac{Ka_3}{M_2(t_j)})^2}-G_3(t_j) \qquad \forall j=0,\dots,119$$

$$\frac{E_1(t_{j+1})-E_1(t_j)}{t_{j+1}-t_j}=\frac{0.1\cdot G_1(t_j)}{K_4+G_1(t_j)}-0.1\cdot E_1(t_j) \qquad \forall j=0,\dots,119$$

$$\frac{E_2(t_{j+1})-E_2(t_j)}{t_{j+1}-t_j}=\frac{0.1\cdot G_2(t_j)}{K_5+G_2(t_j)}-0.1\cdot E_2(t_j) \qquad \forall j=0,\dots,119$$

$$\frac{E_3(t_{j+1})-E_3(t_j)}{t_{j+1}-t_j}=\frac{0.1\cdot G_3(t_j)}{K_6+G_3(t_j)}-0.1\cdot E_3(t_j) \qquad \forall j=0,\dots,119$$

$$\frac{M_1(t_{j+1})-M_1(t_j)}{t_{j+1}-t_j}=\frac{E_1(t_j)\cdot(S-M_1(t_j))}{1+S+M_1(t_j)}-\frac{E_2(t_j)\cdot(M_1(t_j)-M_2(t_j))}{1+M_1(t_j)+M_2(t_j)} \qquad \forall j=0,\dots,119$$

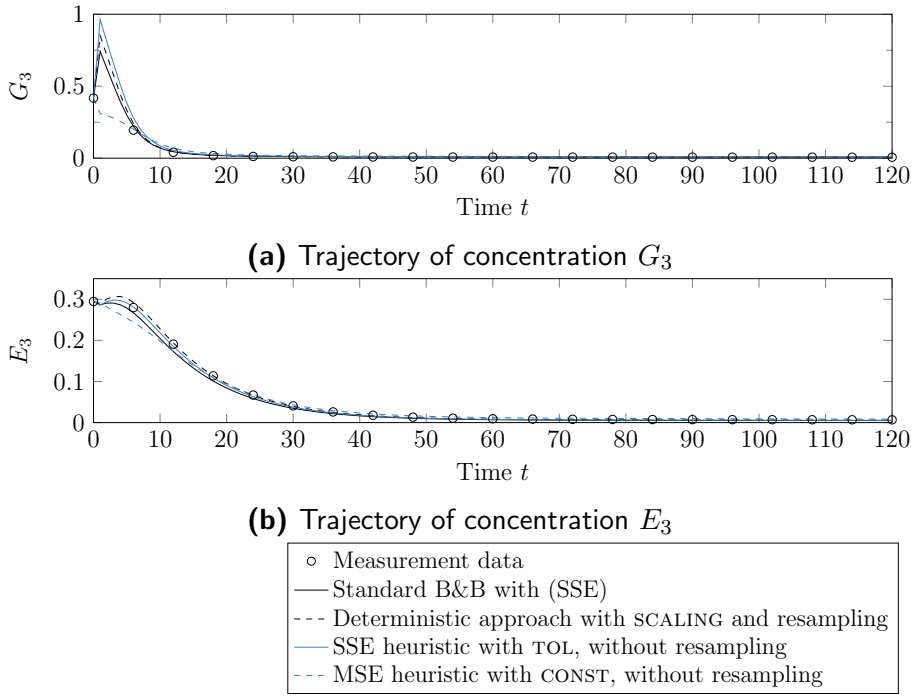
$$\frac{M_2(t_{j+1})-M_2(t_j)}{t_{j+1}-t_j}=\frac{E_2(t_j)\cdot(M_1(t_j)-M_2(t_j))}{1+M_1(t_j)+M_2(t_j)}-\frac{E_3(t_j)\cdot(M_2(t_j)-P)}{1+M_2(t_j)+P} \qquad \forall j=0,\dots,119$$

with initial values at time $t_0 = 0$ as provided in [205], final time $t_f = 120$ and constant step width $\Delta t := t_{j+1} - t_j = 1$ for any $j = 0, \dots, 119$, as well as parameter bounds $Ki_i,\ Ka_i,\ K_{i+3},\ V_i \in [0.05, 100]$ for any $i = 1, 2, 3$. Note that the errors introduced by the

propagation of McCormick relaxations allow for division by zero when optimizing for all degrees of freedom considered in [142, 206] and, thus, result in an infeasible model. Hence, we restrict ourselves to one experiment, namely input values $P = 0.05$ and $S = 0.1$, and reduce the number of unknown parameters. In detail, we fix parameters ni_i, na_i, $kcat_i$, V_{i+3} for any $i = 1, 2, 3$ as well as k_i and Km_i for any $i = 1, \ldots, 6$ to the optimal values reported in Table 2 of [142].
Analogously to [142, 206], we generated synthetic measurements at time points $t_j = 6 \cdot j,\ j = 1, \ldots, 20$. Thus, the full dataset $\mathcal{D}$ of problem TSPexact contains $|\mathcal{D}| = 20$ measurements for each of the 8 differential states. Note that we run 6 Euler steps for each measurement interval to obtain numerically stable integration results yielding 120 discrete time steps in total.
For problem TSPnoisy, we add independent normally distributed errors with standard deviation 0.01 to the exact data $E_{i,d}^{\text{meas}}$, $G_{i,d}^{\text{meas}}$, $M_{i,d}^{\text{meas}}$, $i = 1, 2$ at all time points $t_d = 6 \cdot d,\ d = 0, \ldots, 20$. We fix any negative value obtained to zero since concentrations are non-negative.

D.3.7.2. Optimal solution

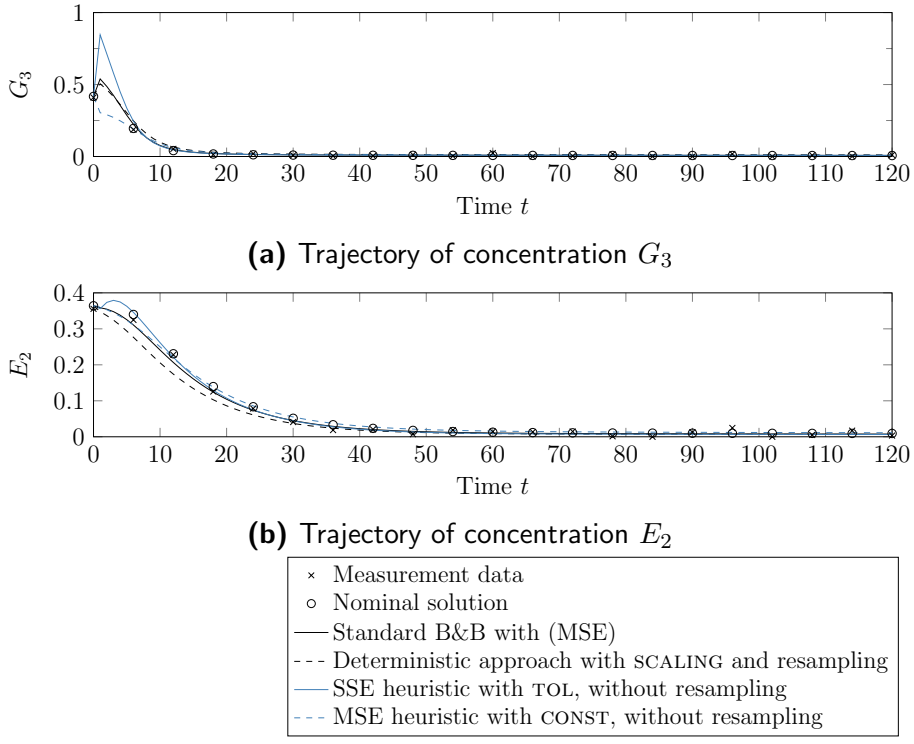
The different approaches determine different solutions for problem TSPexact. Similarly, there are large variations in the corresponding parameter values. In particular, the values of parameters Ki_i, V_i for any $i = 1, 2, 3$ and K_4 are scattered almost over full range $[0.05, 100]$. Only parameters K_5 and K_6 give similar values in range $[1, 2]$ for all approaches. All other parameters values are scattered in range $[0.4, 50]$. As hinted in our previous work [175], the differences between the parameter values mainly affect the first measurement interval $t \in [0, 6]$ of differential states E_2 and G_3, see Figure D.9. The visible differences between the trajectories are between two measurement points and therefore not tracked by the optimization problem. Only the MSE heuristic gives a significant deviation to the measurements of concentration E_3 at $t = 6$. However, we are sure that this deviation lies within the optimality tolerance since we only depict converged solutions in Figure D.9.
For TSPnoisy, we also obtain various different solutions. Only for the MSE heuristic, all augmentation rules lead to the same solution within our CPU time limit of 23 h. Again, the values of parameters Ki_i for any $i = 1, 3$ and V_i for any $i = 1, 2, 3$ are scattered almost over the whole feasible domain $[0.05, 100]$, while only parameter K_6 takes similar values for all approaches, namely $K_6 \in [0.07, 3]$. Remaining parameters Ki_2, Ka_i for any $i = 1, 2, 3$, K_4, and K_5 are scattered over half the range $[0.05, 50]$. For problem TSPnoisy, the trajectories of differential states E_2, E_3, G_2, and G_3 show the largest deviations due to the different parameter values, compare the plots of E_2 and G_3 in Figure D.10 as an example. Similar to the results of problem TSPexact, the solution trajectories for G_3 mainly differ between two measurement points. Thus, these differences cannot be tracked by any optimizer. To obtain a closer fit to the nominal solution of Moles et al. [142], additional measurements in the interval $t \in [0, 6]$ would be required.
Note that we globally solve the TSP model for the specific dataset given. In particular, the optimizer tries to stay close to the data even if these are perturbed due to measurement errors. Thus, trajectories with lower values of concentration E_2 at measurement points $t \in \{6, 18, 30, 36\}$ achieve a better, i.e., smaller, objective value independent of their larger distance to the nominal solution corresponding to the noise-free measurement data, cf. Figure D.10.



(a) Trajectory of concentration G_3

(b) Trajectory of concentration E_3

Figure D.9.: Trajectory of concentrations G_3 and E_3 based on best parameter values found for problem TSPexact by the standard B&B algorithm as well as different approaches of the B&B algorithm with growing datasets compared to the synthetic exact measurement data



(a) Trajectory of concentration G_3

(b) Trajectory of concentration E_2

Figure D.10.: Trajectory of concentrations G_3 and E_2 based on best parameter values found for problem TSPnoisy by the standard B&B algorithm as well as different approaches of the B&B algorithm with growing datasets compared to the synthetic noisy measurement data and the nominal solution, i.e., exact data

Bibliography

[1] R. Abbiw-Jackson, B. Golden, S. Raghavan, and E. Wasil. A divide-and-conquer local search heuristic for data visualization. *Computers & Operations Research*, 33 (11):3070–3087, 2006. doi:10.1016/j.cor.2005.01.020.

[2] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin. *Learning from Data: A Short Course*. AMLBook, 2012. ISBN 9781600490064.

[3] T. Achterberg. SCIP: solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009. doi:10.1007/s12532-008-0001-1.

[4] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas. Global optimization of mixed–integer nonlinear problems. *AIChE Journal*, 46(9):1769–1797, 2000. doi:10.1002/aic.690460908.

[5] C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier. A global optimization method, αBB, for general twice-differentiable constrained NLPs - I. Theoretical advances. *Computers & Chemical Engineering*, 22(9):1137–1158, 1998. doi:10.1016/S0098-1354(98)00027-1.

[6] A. Agosti, P. Ciarletta, H. Garcke, and M. Hinze. Learning patient–specific parameters for a diffuse interface glioblastoma model from neuroimaging data. *Mathematical Methods in the Applied Sciences*, 43(15):8945–8979, 2020. doi:10.1002/mma.6588.

[7] J. Almquist, M. Cvijovic, V. Hatzimanikatis, J. Nielsen, and M. Jirstrand. Kinetic models in industrial biotechnology – Improving cell factory performance. *Metabolic Engineering*, 24:38–60, 2014. doi:10.1016/j.ymben.2014.03.007.

[8] F. Alsmeyer, H.-J. Koss, and W. Marquardt. Indirect spectral hard modeling for the analysis of reactive and interacting mixtures. *Applied Spectroscopy*, 58(8):975–985, 2004. doi:10.1366/0003702041655368.

[9] I. P. Androulakis, C. D. Maranas, and C. A. Floudas. αBB: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization*, 7(4):337–363, 1995. doi:10.1007/BF01099647.

[10] A. C. Atkinson, A. N. Donev, and R. D. Tobias. *Optimum experimental designs, with SAS*, volume 34 of *Oxford statistical science series*. Oxford University Press, Oxford, online edition, 2009. doi:10.1093/oso/9780199296590.001.0001.

[11] E. Augenstein, G. Wrobel, I. Kuperjans, and M. Plessow. Top-energy: computational support for energy system engineering processes. In *First international conference from scientific computing to computational engineering*, 2004.

[12] E. Augenstein, S. Herbergs, I. Kuperjans, and K. Lucas. Simulation of industrial energy supply systems with integrated cost optimization. In *18th International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems*, 2005.

[13] A. Azzalini and A. W. Bowman. A Look at Some Data on the Old Faithful Geyser. *Applied Statistics*, 39(3):357–365, 1990. doi:10.2307/2347385.

[14] F. J. Baader, P. Althaus, A. Bardow, and M. Dahmen. Dynamic ramping for demand response of processes and energy systems based on exact linearization. *Journal of Process Control*, 118:218–230, 2022. doi:10.1016/j.jprocont.2022.08.017.

[15] F. J. Baader, P. Althaus, A. Bardow, and M. Dahmen. Demand response for flat nonlinear MIMO processes using dynamic ramping constraints. *Computers & Chemical Engineering*, 172:108171, 2023. doi:10.1016/j.compchemeng.2023.108171.

[16] A. M. Bagirov, J. Ugon, and H. Mirzayeva. Nonsmooth nonconvex optimization approach to clusterwise linear regression problems. *European Journal of Operational Research*, 229(1):132–142, 2013. doi:10.1016/j.ejor.2013.02.059.

[17] B. Bahl, M. Lampe, P. Voll, and A. Bardow. Optimization-based identification and quantification of demand-side management potential for distributed energy supply systems. *Energy*, 135:889–899, 2017. doi:10.1016/j.energy.2017.06.083.

[18] B. Bahl, T. Söhler, M. Hennen, and A. Bardow. Typical Periods for Two-Stage Synthesis by Time-Series Aggregation with Bounded Error in Objective Function. *Frontiers in Energy Research*, 5:35, 2018. doi:10.3389/fenrg.2017.00035.

[19] J. Banga, C. Moles, and A. Alonso. Global Optimization of Bioprocesses using Stochastic and Hybrid Methods. *Front Global Optimization*, 74, 2003. doi:10.1007/978-1-4613-0251-3_3.

[20] I. H. Bell. Mixture Model for Refrigerant Pairs R-32/1234yf, R-32/1234ze(E), R-1234ze(E)/227ea, R-1234yf/152a, and R-125/1234yf. *Journal of Physical and Chemical Reference Data*, 52(1):013101, 2023. doi:10.1063/5.0135368.

[21] R. Bellman and K. J. Åström. On structural identifiability. *Mathematical Biosciences*, 7(3-4):329–339, 1970. doi:10.1016/0025-5564(70)90132-X.

[22] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Waechter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4-5):597–634, 2009. doi:10.1080/10556780903087124.

[23] T. Benaglia, D. Chauveau, D. R. Hunter, and D. Young. mixtools : An R Package for Analyzing Finite Mixture Models. *Journal of Statistical Software*, 32(6), 2009. doi:10.18637/jss.v032.i06.

[24] P. Beyhaghi, R. Alimo, and T. Bewley. A derivative-free optimization algorithm for the efficient minimization of functions obtained via statistical averaging. *Computational Optimization and Applications*, 76(1):1–31, 2020. doi:10.1007/s10589-020-00172-4.

[25] S. K. Bharti and R. Roy. Quantitative 1H NMR spectroscopy. *TrAC Trends in Analytical Chemistry*, 35:5–26, 2012. doi:10.1016/j.trac.2012.02.007.

[26] L. T. Biegler and C. A. Jacobson. Optimization Opportunities in Product Development: Perspective from A Manufacturing Company. In S. G. Munoz, C. D. Laird, and M. J. Realff, editors, *FOCAPD-19/Proceedings of the 9th International Conference on Foundations of Computer-Aided Process Design, July 14 - 18 2019*, volume 47 of *Computer Aided Chemical Engineering*, pages 275–286. Elsevier, San Diego and Oxford, 2019. doi:10.1016/B978-0-12-818597-1.50044-8.

[27] H. G. Bock and K.-J. Plitt. A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems. *IFAC Proceedings Volumes*, 17(2):1603–1608, 1984. doi:10.1016/S1474-6670(17)61205-9.

[28] H. G. Bock, E. Kostina, and J. P. Schlöder. Numerical Methods for Parameter Estimation in Nonlinear Differential Algebraic Equations. *GAMM-Mitteilungen*, 30 (2):376–408, 2007. doi:10.1002/gamm.200790024.

[29] E. Bogar. Chaos Game Optimization-Least Squares Algorithm for Photovoltaic Parameter Estimation. *Arabian Journal for Science and Engineering*, 48(5):6321–6340, 2023. doi:10.1007/s13369-022-07364-6.

[30] G. M. Bollas, P. I. Barton, and A. Mitsos. Bilevel Optimization Formulation for Parameter Estimation in Vapor-Liquid(-Liquid) Phase Equilibrium Problems. *Chemical Engineering Science*, 64(8):1768–1783, 2009. doi:10.1016/j.ces.2009.01.003.

[31] D. Bongartz, J. Najman, S. Sass, and A. Mitsos. MAiNGO – McCormick-based Algorithm for mixed-integer Nonlinear Global Optimization. Process Systems Engineering (AVT.SVT), RWTH Aachen University, 2018. URL `http://permalink.avt.rwth-aachen.de/?id=729717`, Last accessed: Dec 1, 2024.

[32] D. Bongartz and A. Mitsos. Deterministic global optimization of process flowsheets in a reduced space using McCormick relaxations. *Journal of Global Optimization*, 69 (4):761–796, 2017. doi:10.1007/s10898-017-0547-4.

[33] F. Boukouvala, R. Misener, and C. A. Floudas. Global optimization advances in Mixed-Integer Nonlinear Programming, MINLP, and Constrained Derivative-Free Optimization, CDFO. *European Journal of Operational Research*, 252(3):701–727, 2016. doi:10.1016/j.ejor.2015.12.018.

[34] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. doi:10.1007/BF00058655.

[35] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi:10.1023/A:1010933404324.

[36] R. G. Brusch and R. H. Schappelle. Solution of Highly Constrained Optimal Control Problems Using Nonlinear Programing. *AIAA Journal*, 11(2):135–136, 1973.

[37] M. R. Bussieck, A. S. Drud, and A. Meeraus. MINLPLib—A Collection of Test Models for Mixed-Integer Nonlinear Programming. *INFORMS Journal on Computing*, 15(1):114–119, 2003. doi:10.1287/ijoc.15.1.114.15159.

[38] R. H. Byrd, J. Nocedal, and R. A. Waltz. Knitro: An Integrated Package for Nonlinear Optimization. In *Large-scale nonlinear optimization*, pages 35–59. Springer, 2006. doi:10.1007/0-387-30065-1_4.

[39] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer. A stochastic quasi-Newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016. doi:10.1137/140954362.

[40] F. Casella and A. Leva. Modelling of thermo-hydraulic power generation processes using Modelica. *Mathematical and Computer Modelling of Dynamical Systems*, 12 (1):19–33, 2006. doi:10.1080/13873950500071082.

[41] A. Caspari, A. M. Bremen, J. M. Faust, F. Jung, C. D. Kappatou, S. Sass, Y. Vaupel, R. Hannemann-Tamás, A. Mhamdi, and A. Mitsos. DyOS - A Framework for Optimization of Large-Scale Differential Algebraic Equation Systems. In Anton A. Kiss, Edwin Zondervan, Richard Lakerveld, and Leyla Özkan, editors, *29th European Symposium on Computer Aided Process Engineering*, volume 46 of *Computer Aided Chemical Engineering*, pages 619–624. Elsevier, 2019. doi:10.1016/B978-0-12-818634-3.50104-1.

[42] B. Chachuat, A. B. Singer, and P. I. Barton. Global mixed-integer dynamic optimization. *AIChE Journal*, 51(8):2235–2253, 2005. doi:10.1002/aic.10494.

[43] B. Chachuat, B. Houska, R. Paulen, N. Perić, J. Rajyaguru, and M. E. Villanueva. Set-Theoretic Approaches in Analysis, Estimation and Control of Nonlinear Systems. *IFAC-PapersOnLine*, 48(8):981–995, 2015. doi:10.1016/j.ifacol.2015.09.097. URL `https://github.com/omega-icl/mcpp`, Last accessed: Nov 11, 2019.

[44] R. A. Cox. Comparison of the performance of seven optimization algorithms on twelve unconstrained optimization problems. ref. 1335cn04. *Gulf Research and Development Company, Pittsburg*, 1969.

[45] A. D'Amato, Y. Wang, D. Filev, and E. Remes. On the Tradeoffs between Static and Dynamic Adaptive Optimization for an Automotive Application. *SAE International Journal of Commercial Vehicles*, 10(1):346–352, 2017. doi:10.4271/2017-01-0605.

[46] J. Deboever and S. Grijalva. Modeling and optimal scheduling of integrated thermal and electrical energy microgrid. In *North American Power Symposium (NAPS), 2016*, pages 1–6, 2016. doi:10.1109/NAPS.2016.7747841.

[47] C. D. Demirhan, W. W. Tso, G. S. Ogumerem, and E. N. Pistikopoulos. Energy systems engineering - a guided tour. *BMC Chemical Engineering*, 1(1), 2019. doi:10.1186/s42480-019-0009-5.

[48] C. D. Demirhan, W. W. Tso, J. B. Powell, C. F. Heuberger, and E. N. Pistikopoulos. A Multiscale Energy Systems Engineering Approach for Renewable Power Generation and Storage Optimization. *Industrial & Engineering Chemistry Research*, 59 (16):7706–7721, 2020. doi:10.1021/acs.iecr.0c00436.

[49] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 39(1):1–38, 1977. doi:10.1111/j.2517-6161.1977.tb01600.x.

[50] S. Deutz, D. Bongartz, B. Heuser, A. Kätelhön, L. Schulze-Langenhorst, A. Omari, M. Walters, J. Klankermayer, W. Leitner, A. Mitsos, S. Pischinger, and A. Bardow. Cleaner production of cleaner fuels: wind-to-wheel–environmental assessment of CO 2-based oxymethylene ether as a drop-in fuel. *Energy & Environmental Science*, 11 (2):331–343, 2018. doi:10.1039/C7EE01657C.

[51] H. Diedam and S. Sager. Global optimal control with the direct multiple shooting method. *Optimal Control Applications and Methods*, 39(2):449–470, 2018. doi:10.1002/oca.2324.

[52] H. Diedam. *Global optimal control using direct multiple shooting*. Dissertation, Ruprecht-Karls-Universität, Heidelberg, 2015. doi:10.11588/heidok.00019744.

[53] J. Du, J. Park, I. Harjunkoski, and M. Baldea. A time scale-bridging approach for integrating production scheduling and process control. *Computers & Chemical Engineering*, 79:59–69, 2015. doi:10.1016/j.compchemeng.2015.04.026.

[54] K. Du and R. B. Kearfott. The cluster problem in multivariate global optimization. *Journal of Global Optimization*, 5(3):253–265, 1994. doi:10.1007/BF01096455.

[55] DWD Climate Data Center (CDC). Historical hourly station observations from 2018, version v006, 2018. URL `https://cdc.dwd.de/portal/`, Last accessed: Jul 24, 2019.

[56] A. Echtermeyer, C. Marks, A. Mitsos, and J. Viell. Inline Raman Spectroscopy and Indirect Hard Modeling for Concentration Monitoring of Dissociated Acid Species. *Applied Spectroscopy*, 75(5):506–519, 2021. doi:10.1177/0003702820973275.

[57] A. W. W. Echtermeyer, C. Marks, A. Mitsos, and J. Viell. Dataset to "Inline Raman Spectroscopy and Indirect Hard Modeling for Concentration Monitoring of Dissociated Acid Species", 2024. doi:10.18154/RWTH-2024-01177.

[58] B. Efron and R. Tibshirani. *An introduction to the bootstrap*, volume 57 of *Monographs on statistics and applied probability*. Chapman & Hall/CRC, New York, 1994. doi:10.1201/9780429246593.

[59] J. A. Egea, D. Henriques, T. Cokelaer, A. F. Villaverde, A. MacNamara, D.-P. Danciu, J. R. Banga, and J. Saez-Rodriguez. MEIGO: an open-source software suite based on metaheuristics for global optimization in systems biology and bioinformatics. *BMC Bioinformatics*, 15, 2014. doi:10.1186/1471-2105-15-136.

[60] W. R. Esposito and C. A. Floudas. Global Optimization for the Parameter Estimation of Differential-Algebraic Systems. *Industrial & Engineering Chemistry Research*, 39(5):1291–1310, 2000. doi:10.1021/ie990486w.

[61] E. Fabrizio. *Modelling of multi-energy systems in buildings*. Dissertation, Politecnico di Torino and Institut National des Sciences Appliquées de Lyon, 2008.

[62] H. Fanaee-T. Bike Sharing Dataset, 2013. doi:10.24432/C5W894.

[63] H. Fanaee-T and J. Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2(2-3):113–127, 2014. doi:10.1007/s13748-013-0040-3.

[64] S. Fazlollahi, N. Schüler, and F. Maréchal. A solid thermal storage model for the optimization of buildings operation strategy. *Energy*, 88:209–222, 2015. doi:10.1016/j.energy.2015.04.085.

[65] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. doi:10.1145/358669.358692.

[66] A. Flores-Tlacuahuac and L. T. Biegler. Simultaneous mixed-integer dynamic optimization for integrated design and control. *Computers & Chemical Engineering*, 31 (5–6):588–600, 2007. doi:10.1016/j.compchemeng.2006.08.010.

[67] C. A. Floudas. *Deterministic global optimization: Theory, Methods and Applications*, volume 37. Springer, New York, NY, 2000. doi:10.1007/978-1-4757-4949-6.

[68] J. J. Forrest, S. Vigerske, T. Ralphs, L. Hafer, J. P. Fasano, H. G. Santos, M. Saltzman, H. Gassmann, B. Kristjansson, and A. King. COIN-OR Linear Programming Solver. *COIN-OR Foundation*, 2019. URL `https://github.com/coin-or/Clp`, Last accessed: Nov 8, 2019.

[69] J. Forrest, D. de La Nuez, and R. Lougee-Heimer. CLP User Guide. IBM Corporation, 2004.

[70] S. S. Foslie, B. R. Knudsen, and M. Korpås. Integrated design and operational optimization of energy systems in dairies. *Energy*, 281:128242, 2023. doi:10.1016/j.energy.2023.128242.

[71] C. A. Frangopoulos. Recent developments and trends in optimization of energy systems. *Energy*, 164:1011–1020, 2018. doi:10.1016/j.energy.2018.08.218.

[72] J. Fu, J. M. Faust, B. Chachuat, and A. Mitsos. Local optimization of dynamic programs with guaranteed satisfaction of path constraints. *Automatica*, 62:184–192, 2015. doi:10.1016/j.automatica.2015.09.013.

[73] GAMS Development Corporation. General Algebraic Modeling System (GAMS) Release 25.1.1, 2018. URL `https://www.gams.com/download/`, Last accessed: Dec 1, 2024.

[74] D. J. Gardner, D. R. Reynolds, C. S. Woodward, and C. J. Balos. Enabling New Flexibility in the SUNDIALS Suite of Nonlinear and Differential/Algebraic Equation Solvers. *ACM Transactions on Mathematical Software*, 48(3):1–24, 2022. doi:10.1145/3539801.

[75] C. Y. Gau and M. A. Stadtherr. Deterministic global optimization for error-in-variables parameter estimation. *AIChE Journal*, 48(6):1192–1197, 2002. doi:10.1002/aic.690480607.

[76] C. F. Gauss. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*. Sumtibus F. Perthes et I.H. Besser, 1809.

[77] A. Ghobeity and A. Mitsos. Optimal Design and Operation of a Solar Energy Receiver and Storage. *Journal of Solar Energy Engineering*, 134(3):031005, 2012. doi:10.1115/1.4006402.

[78] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005. doi:10.1137/s0036144504446096.

[79] A. M. Gleixner, T. Berthold, B. Müller, and S. Weltge. Three enhancements for optimization-based bound tightening. *Journal of Global Optimization*, 67(4):731–757, 2017. doi:10.1007/s10898-016-0450-4.

[80] F. Glover and G. A. Kochenberger. *Handbook of metaheuristics*, volume 57 of *International series in operations research & management science*. Kluwer, Boston, 2003. doi:10.1007/b101874.

[81] S. Goderbauer, B. Bahl, P. Voll, M. E. Lübbecke, A. Bardow, and A. M. Koster. An adaptive discretization MINLP algorithm for optimal synthesis of decentralized energy supply systems. *Computers & Chemical Engineering*, 95:38–48, 2016. doi:10.1016/j.compchemeng.2016.09.008.

[82] T. Goerke and S. Engell. Application of evolutionary algorithms in guaranteed parameter estimation. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 5100–5105, 2016. doi:10.1109/CEC.2016.7748336.

[83] P. Goos and B. Jones. *Optimal design of experiments: A case study approach*. Wiley, Chichester, 2011. doi:10.1002/9781119974017.

[84] I. E. Grossmann, J. A. Caballero, and H. Yeomans. Advances in mathematical programming for the synthesis of process systems. *Latin American Applied Research*, 30(4):263–284, 2000.

[85] S. Gunasekaran, N. D. Mancini, and A. Mitsos. Optimal design and operation of membrane-based oxy-combustion power plants. *Energy*, 70:338–354, 2014. doi:10.1016/j.energy.2014.04.008.

[86] A. Gupta, A. Kumar, M. P'al, and T. Roughgarden. Approximation via cost sharing. *Journal of the ACM*, 54(3):11, 2007. doi:10.1145/1236457.1236458.

[87] R. Hannemann, W. Marquardt, U. Naumann, and B. Gendler. Discrete first- and second-order adjoints and automatic differentiation for the sensitivity analysis of dynamic models. *Procedia Computer Science*, 1(1):297–305, 2010. doi:10.1016/j.procs.2010.04.033.

[88] W. Härdle. *Smoothing techniques: With implementation in S*. Springer series in statistics. Springer, New York, NY, 1991. doi:10.1007/978-1-4612-4432-5.

[89] H. W. Häring. *Industrial gases processing*. Wiley-VCH Verlag, Weinheim, 2008. doi:10.1002/9783527621248.

[90] F. E. Harrell, K. L. Lee, R. M. Califf, D. B. Pryor, and R. A. Rosati. Regression modelling strategies for improved prognostic prediction. *Statistics in Medicine*, 3(2): 143–152, 1984. doi:10.1002/sim.4780030207.

[91] F. E. Harrell, K. L. Lee, and D. B. Mark. Multivariable prognostic models: Issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in Medicine*, 15(4):361–387, 1996. doi:10.1002/(SICI)1097-0258(19960229)15:4<361::AID-SIM168>3.0.CO;2-4.

[92] M. A. Hassan, S. Serra, S. Sochard, H. Viot, F. Marias, and J.-M. Reneaume. Holistic non-linear optimization of the layout, sizing, and operation of a district heating plant. *Energy Conversion and Management*, 301:118079, 2024. doi:10.1016/j.enconman.2024.118079.

[93] T. J. Hastie, J. H. Friedman, and R. Tibshirani. *The elements of statistical learning: Data mining, inference, and prediction*. Springer, New York, NY, 2 edition, 2009. doi:10.1007/978-0-387-84858-7.

[94] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. SUNDIALS. *ACM Transactions on Mathematical Software*, 31(3):363–396, 2005. doi:10.1145/1089014.1089020.

[95] G. E. Hinton. Learning translation invariant recognition in a massively parallel networks. In G. Goos, J. Hartmanis, D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries, D. Luckham, C. Moler, A. Pnueli, G. Seegmüller, J. Stoer, N. Wirth, J. W. Bakker, A. J. Nijman, and P. C. Treleaven, editors, *PARLE Parallel Architectures and Languages Europe*, volume 258 of *Lecture Notes in Computer Science*, pages 1–13. Springer, Berlin, Heidelberg, 1987. doi:10.1007/3-540-17943-7_117.

[96] R. Horst and P. M. Pardalos. *Handbook of global optimization*, volume 2. Springer, New York, NY, 1995. doi:10.1007/978-1-4615-2025-2.

[97] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer, Berlin, Heidelberg, 3 edition, 1996. doi:10.1007/978-3-662-03199-5.

[98] B. Houska and B. Chachuat. Branch-and-Lift Algorithm for Deterministic Global Optimization in Nonlinear Optimal Control. *Journal of Optimization Theory and Applications*, 162(1):208–248, 2014. doi:10.1007/s10957-013-0426-1.

[99] P. J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. doi:10.1214/aoms/1177703732.

[100] P. J. Huber. The place of the L1-norm in robust estimation. *Computational Statistics & Data Analysis*, 5(4):255–262, 1987. doi:10.1016/0167-9473(87)90049-1.

[101] International Business Machines Corporation. IBM ILOG CPLEX Optimization Studio 22.1.1, 2022.

[102] G. James, T. Hastie, D. Witten, and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R*, volume 103. Springer, New York, NY, 2013. doi:10.1007/978-1-4614-7138-7.

[103] S. G. Johnson. The NLopt nonlinear-optimization package, 2007. URL `http://github.com/stevengj/nlopt`, Last accessed: Jan 21, 2021.

[104] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13:455–492, 1998. doi:10.1023/A:1008306431147.

[105] P. Joy, H. Djelassi, A. Mhamdi, and A. Mitsos. Optimization-based global structural identifiability. *Short Communication; Computers & Chemical Engineering*, 128:417–420, 2019. doi:10.1016/j.compchemeng.2019.06.019.

[106] M. N. Jung, C. Kirches, S. Sager, and S. Sass. Computational Approaches for Mixed Integer Optimal Control Problems with Indicator Constraints. *Vietnam Journal of Mathematics*, 46(4):1023–1051, 2018. doi:10.1007/s10013-018-0313-z.

[107] E.-M. Kapfer, P. Stapor, and J. Hasenauer. Challenges in the calibration of large-scale ordinary differential equation models. *IFAC-PapersOnLine*, 52(26):58–64, 2019. doi:10.1016/j.ifacol.2019.12.236.

[108] C. D. Kappatou, D. Bongartz, J. Najman, S. Sass, and A. Mitsos. Global dynamic optimization with Hammerstein-Wiener models embedded. *Journal of Global Optimization*, 84(2):321–347, 2022. doi:10.1007/s10898-022-01145-z.

[109] J. E. Kelley. The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, 8(4):703–712, 1960. doi:10.1137/0108053.

[110] M. T. Kelley, R. C. Pattison, R. Baldick, and M. Baldea. An efficient MILP framework for integrating nonlinear process dynamics and control in optimal production scheduling calculations. *Computers & Chemical Engineering*, 110:35–52, 2018. doi:10.1016/j.compchemeng.2017.11.021.

[111] A. Khajavirad and N. V. Sahinidis. A hybrid LP/NLP paradigm for global optimization relaxations. *Mathematical Programming Computation*, 10(3):383–421, 2018. doi:10.1007/s12532-018-0138-5.

[112] P. Kirst, O. Stein, and P. Steuermann. Deterministic upper bounds for spatial branch-and-bound methods in global minimization with nonconvex constraints. *TOP*, 23(2):591–616, 2015. doi:10.1007/s11750-015-0387-7.

[113] R. W. Koller, L. A. Ricardez-Sandoval, and L. T. Biegler. Stochastic back-off algorithm for simultaneous design, control, and scheduling of multiproduct systems under uncertainty. *AIChE Journal*, 64(7):2379–2389, 2018. doi:10.1002/aic.16092.

[114] D. Kovačević, N. Mladenović, B. Petrović, and P. Milošević. DE-VNS: Self-adaptive Differential Evolution with crossover neighborhood search for continuous global optimization. *Computers & Operations Research*, 52:157–169, 2014. doi:10.1016/j.cor.2013.12.009.

[115] D. Kraft. On Converting Optimal Control Problems into Nonlinear Programming Problems. In K. Schittkowski, editor, *Computational Mathematical Programming*, NATO ASI series, pages 261–280, Berlin, Heidelberg, 1985. Springer. doi:10.1007/978-3-642-82450-0_9.

[116] E. Kriesten, F. Alsmeyer, A. Bardow, and W. Marquardt. Fully automated indirect hard modeling of mixture spectra. *Chemometrics and Intelligent Laboratory Systems*, 91(2):181–193, 2008. doi:10.1016/j.chemolab.2007.11.004.

[117] Y. Kristianto and A. Gunasekaran. A global optimization for sustainable multi-domain global manufacturing. *Computers & Operations Research*, 89:307–323, 2018. doi:10.1016/j.cor.2015.12.001.

[118] U. Kulisch. *C++ Toolbox for Verified Computing I: Basic Numerical Problems Theory, Algorithms, and Programs*. Springer eBook Collection Mathematics and Statistics. Springer, Berlin, Heidelberg, 1995. doi:10.1007/978-3-642-79651-7.

[119] H. J. Kushner. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering*, 86(1): 97–106, 1964. doi:10.1115/1.3653121.

[120] C. L. Lara, D. S. Mallapragada, D. J. Papageorgiou, A. Venkatesh, and I. E. Grossmann. Deterministic electric power infrastructure planning: Mixed-integer programming model and nested decomposition algorithm. *European Journal of Operational Research*, 271(3):1037–1054, 2018. doi:10.1016/j.ejor.2018.05.039.

[121] A. C. Lazanas and M. I. Prodromidis. Electrochemical Impedance Spectroscopy - A Tutorial. *ACS measurement science au*, 3(3):162–193, 2023. doi:10.1021/acsmeasuresciau.2c00070.

[122] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient BackProp. In G. Montavon, G. B. Orr, and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade*, volume 7700 of *Lecture Notes in Computer Science*, pages 9–48. Springer, Berlin, Heidelberg, 2012. doi:10.1007/978-3-642-35289-8_3.

[123] E. W. Lemmon and R. T. Jacobsen. A New Functional Form and New Fitting Techniques for Equations of State with Application to Pentafluoroethane (HFC-125). *Journal of Physical and Chemical Reference Data*, 34(1):69–108, 2005. doi:10.1063/1.1797813.

[124] E. W. Lemmon, M. O. McLinden, and W. Wagner. 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. *Journal of Chemical & Engineering Data*, 54(12):3141–3180, 2009. doi:10.1021/je900217v.

[125] M. Lerch, G. Tischler, J. W. von Gudenberg, W. Hofschuster, and W. Krämer. FILIB++, a fast interval library supporting containment computations. *ACM Transactions on Mathematical Software*, 32(2):299–324, 2006. doi:10.1145/1141885.1141893.

[126] C. Li, D. E. Bernal, K. C. Furman, M. A. Duran, and I. E. Grossmann. Sample average approximation for stochastic nonconvex mixed integer nonlinear programming via outer-approximation. *Optimization and Engineering*, 22(3):1245–1273, 2021. doi:10.1007/s11081-020-09563-2.

[127] X. Li and P. I. Barton. Optimal design and operation of energy systems under uncertainty. *Journal of Process Control*, 30:1–9, 2015. doi:10.1016/j.jprocont.2014.11.004.

[128] F. Lin, S. Leyffer, and T. Munson. A two-level approach to large mixed-integer programs with application to cogeneration in energy-efficient buildings. *Computational Optimization and Applications*, 65(1):1–46, 2016. doi:10.1007/s10589-016-9842-0.

[129] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989. doi:10.1007/BF01589116.

[130] E. Lizarraga-Garcia, A. Ghobeity, M. Totten, and A. Mitsos. Optimal operation of a solar-thermal power plant with energy storage and electricity buy-back from grid. *Energy*, 51:61–70, 2013. doi:10.1016/j.energy.2013.01.024.

[131] M. Locatelli and F. Schoen. *Global optimization: Theory, Algorithms, and Applications*, volume 15. MOS-SIAM series on optimization, 2013. doi:10.1137/1.9781611972672.

[132] L. Maier, D. Jansen, F. Wüllhorst, M. Kremer, A. Kümpel, T. Blacha, and D. Müller. AixLib: an open-source Modelica library for compound building energy systems from component to district level with automated quality management. *Journal of Building Performance Simulation*, pages 1–24, 2023. doi:10.1080/19401493.2023.2250521.

[133] D. E. Majewski, M. Lampe, P. Voll, and A. Bardow. TRusT: A Two-stage Robustness Trade-off approach for the design of decentralized energy supply systems. *Energy*, pages –, 2016. doi:10.1016/j.energy.2016.10.065.

[134] A. Malcolm, J. Polan, L. Zhang, B. A. Ogunnaike, and A. A. Linninger. Integrating systems design and control using dynamic flexibility analysis. *AIChE Journal*, 53(8): 2048–2061, 2007. doi:10.1002/aic.11218.

[135] F. Malz and H. Jancke. Validation of quantitative NMR. *Journal of Pharmaceutical and Biomedical Analysis*, 38(5):813–823, 2005. doi:10.1016/j.jpba.2005.01.043.

[136] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I – Convex underestimating problems. *Mathematical Programming*, 10 (1):147–175, 1976. doi:10.1007/BF01580665.

[137] F. Messerer, K. Baumgärtner, and M. Diehl. Survey of sequential convex programming and generalized Gauss-Newton methods. *ESAIM: Proceedings and Surveys*, 71: 64–88, 2021. doi:10.1051/proc/202171107.

[138] A. Miro, C. Pozo, G. Guillen-Gosalbez, J. A. Egea, and L. Jimenez. Deterministic global optimization algorithm based on outer approximation for the parameter estimation of nonlinear dynamic biological systems. *BMC Bioinformatics*, 13, 2012. doi:10.1186/1471-2105-13-90.

[139] R. Misener and C. Floudas. ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. *Journal of Global Optimization*, 59: 503–526, 2014. doi:10.1007/s10898-014-0166-2.

[140] A. Mitsos, B. Chachuat, and P. I. Barton. McCormick-Based Relaxations of Algorithms. *SIAM Journal on Optimization*, 20(2):573–601, 2009. doi:10.1137/080717341.

[141] J. Močkus. On bayesian methods for seeking the extremum. In G. Goos, J. Hartmanis, P. Brinch Hansen, D. Gries, C. Moler, G. Seegmüller, N. Wirth, and G. I. Marchuk, editors, *Optimization techniques*, volume 27 of *Lecture Notes in Control and Information Sciences*, pages 400–404. Springer, Berlin, Heidelberg, 1975. doi:10.1007/3-540-07165-2_55.

[142] C. G. Moles, P. Mendes, and J. R. Banga. Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome research*, 13(11): 2467–2474, 2003. doi:10.1101/gr.1262503.

[143] J. J. Moré, B. S. Garbow, and K. E. Hillstrom. Testing Unconstrained Optimization Software. *ACM Transactions on Mathematical Software*, 7(1):17–41, 1981. doi:10.1145/355934.355936.

[144] K. P. Murphy. *Probabilistic machine Learning: An introduction*. Adaptive computation and machine learning. The MIT press, Cambridge, Massachusetts, 2022. ISBN 9780262369305.

[145] A. Murray, T. Faulwasser, and V. Hagenmeyer. Mixed-Integer vs. Real-Valued Formulations of Battery Scheduling Problems. *IFAC-PapersOnLine*, 51(28):350–355, 2018. doi:10.1016/j.ifacol.2018.11.727.

[146] J. Najman, D. Bongartz, A. Tsoukalas, and A. Mitsos. Erratum to: Multivariate McCormick relaxations. *Journal of Global Optimization*, 68(1):219–225, 2017. doi:10.1007/s10898-016-0470-0.

[147] J. Najman, D. Bongartz, and A. Mitsos. Linearization of McCormick relaxations and hybridization with the auxiliary variable method. *Journal of Global Optimization*, 80(4):731–756, 2021. doi:10.1007/s10898-020-00977-x.

[148] J. Nocedal. Updating Quasi-Newton Matrices with Limited Storage. *Mathematics of Computation*, 35(151):773–782, 1980. doi:10.2307/2006193.

[149] B. A. Ogunnaike and W. H. Ray. *Process dynamics, modeling, and control*, volume 1. Oxford University Press New York, 1994. ISBN 9780195091199.

[150] M. R. Osborne. Some aspects of nonlinear least squares calculations. In F. A. Lootsma, editor, *Numerical methods for nonlinear optimization*, pages 171–189. Academic Press, New York, 1972.

[151] X. Pan, N. Geng, and X. Xie. Appointment scheduling and real-time sequencing strategies for patient unpunctuality. *European Journal of Operational Research*, 295 (1):246–260, 2021. doi:10.1016/j.ejor.2021.02.055.

[152] I. Papamichail and C. Adjiman. A Rigorous Global Optimization Algorithm for Problems with Ordinary Differential Equations. *Journal of Global Optimization*, 24: 1–33, 2002. doi:10.1023/A:1016259507911.

[153] P. M. Pardalos and H. E. Romeijn. *Handbook of global optimization*, volume 62 of *Nonconvex optimization and its applications*. Springer, New York and London, 2011. doi:10.1007/978-1-4757-5362-2.

[154] P. Peduzzi, J. Concato, E. Kemper, T. R. Holford, and A. R. Feinstein. A simulation study of the number of events per variable in logistic regression analysis. *Journal of Clinical Epidemiology*, 49(12):1373–1379, 1996. doi:10.1016/s0895-4356(96)00236-3.

[155] J. R. Pillai, K. Heussen, and P. A. Østergaard. Comparative analysis of hourly and dynamic power balancing models for validating future energy scenarios. *Energy*, 36 (5):3233–3243, 2011. doi:10.1016/j.energy.2011.03.014.

[156] E. N. Pistikopoulos and Y. Tian. Advanced Modeling and Optimization Strategies for Process Synthesis. *Annual Review of Chemical and Biomolecular Engineering*, 2024. doi:10.1146/annurev-chembioeng-100522-112139.

[157] J. Puschke, A. Zubov, J. Kosek, and A. Mitsos. Multi-model approach based on parametric sensitivities – A heuristic approximation for dynamic optimization of semi-batch processes with parametric uncertainties. *Computers & Chemical Engineering*, 98:161–179, 2017. doi:10.1016/j.compchemeng.2016.12.004.

[158] Y. Qin, P. Liu, and Z. Li. Multi-timescale hierarchical scheduling of an integrated energy system considering system inertia. *Renewable and Sustainable Energy Reviews*, 169:112911, 2022. doi:10.1016/j.rser.2022.112911.

[159] M. H. Quenouille. Notes on Bias in Estimation. *Biometrika*, 43(3/4):353, 1956. doi:10.2307/2332914.

[160] G. C. V. Ramadas, Fernandes, Edite M. G. P., A. M. V. Ramadas, Rocha, Ana Maria A. C., and M. F. P. Costa. On Metaheuristics for Solving the Parameter Estimation Problem in Dynamic Systems: A Comparative Study. *Journal of Optimization*, 2018. doi:10.1155/2018/3213484.

[161] J. R. Rice and J. S. White. Norms for Smoothing and Estimation. *SIAM Review*, 6(3):243–256, 1964. URL http://www.jstor.org/stable/2027590, Last accessed: Dec 1, 2024.

[162] C. Richter. *Proposal of New Object-Oriented Equation-Based Model Libraries for Thermodynamic Systems*. Dissertation, Technische Universität Braunschweig, 2008. doi:10.24355/dbbs.084-200806100200-3.

[163] J. P. A. S. Rocha, F. S. Loureiro, H. S. Bernardino, A. B. Vieira, and H. J. C. Barbosa. Metaheuristics applied to the thermographic detection of multicentric breast tumor. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 46(6), 2024. doi:10.1007/s40430-024-04907-w.

[164] P. J. Rousseeuw. Least Median of Squares Regression. *Journal of the American Statistical Association*, 79(388):871, 1984. doi:10.2307/2288718.

[165] D. Rulliere, A. Faleh, F. Planchet, and W. Youssef. Exploring or reducing noise? A global optimization algorithm in the presence of noise. *Structural and Multidisciplinary Optimization*, 47(6):921–936, 2013. doi:10.1007/s00158-012-0874-5.

[166] H. S. Ryoo and N. V. Sahinidis. Global optimization of nonconvex NLPs and MINLPs with applications in process design. *Computers & Chemical Engineering*, 19(5):551–566, 1995. doi:10.1016/0098-1354(94)00097-2.

[167] H. S. Ryoo and N. V. Sahinidis. A branch-and-reduce approach to global optimization. *Journal of Global Optimization*, 8(2):107–138, 1996. doi:10.1007/BF00138689.

[168] S-PACT GmbH. PEAXACT User Manual: Version 5.7. Aachen, Germany, 2022.

[169] R. W. H. Sargent and G. R. Sullivan. The development of an efficient optimal control package. In J. Stoer, editor, *Optimization Techniques*, volume 7 of *Lecture Notes in Control and Information Sciences*, pages 158–168. Springer, Berlin, Heidelberg, 1978. doi:10.1007/BFb0006520.

[170] S. Sass and A. Mitsos. Optimal Operation of Dynamic (Energy) Systems: When are Quasi-Steady Models Adequate? *Computers & Chemical Engineering*, 124:133–139, 2019. doi:10.1016/j.compchemeng.2019.02.011.

[171] S. Sass and J. R. Seidenberg. GloPSE/ElectrochemicalImpedanceSpectroscopy2CPE, 2024. URL https://git.rwth-aachen.de/avt-svt/public/glopse, Last accessed: Dec 1, 2024.

[172] S. Sass, T. Faulwasser, D. E. Hollermann, C. D. Kappatou, D. Sauer, T. Schütz, D. Y. Shu, A. Bardow, L. Gröll, V. Hagenmeyer, D. Müller, and A. Mitsos. Model compendium, data, and optimization benchmarks for sector-coupled energy systems. *Computers & Chemical Engineering*, 135:106760, 2020. doi:10.1016/j.compchemeng.2020.106760.

[173] S. Sass, A. Tsoukalas, I. H. Bell, D. Bongartz, J. Najman, and A. Mitsos. Dataset from: Towards Global Parameter Estimation Exploiting Reduced Data Sets, 2022. doi:10.18154/RWTH-2022-03252.

[174] S. Sass, A. Tsoukalas, I. H. Bell, D. Bongartz, J. Najman, and A. Mitsos. Towards global parameter estimation exploiting reduced data sets. *Optimization Methods and Software*, 38(6):1129–1141, 2023. doi:10.1080/10556788.2023.2205645.

[175] S. Sass, A. Mitsos, D. Bongartz, I. H. Bell, N. I. Nikolov, and A. Tsoukalas. A branch-and-bound algorithm with growing datasets for large-scale parameter estimation. *European Journal of Operational Research*, 316(1):36–45, 2024. doi:10.1016/j.ejor.2024.02.020.

[176] S. Sass, A. Mitsos, N. I. Nikolov, and A. Tsoukalas. Out-of-sample estimation for a branch-and-bound algorithm with growing datasets. *Submitted*, 2024.

[177] S. Schemme, R. C. Samsun, R. Peters, and D. Stolten. Power-to-fuel as a key to sustainable transport systems – An analysis of diesel fuels produced from CO2 and renewable electricity. *Fuel*, 205:198–221, 2017. doi:10.1016/j.fuel.2017.05.061.

[178] N. N. Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7):1723–1738, 2002. doi:10.1162/08997660260028683.

[179] T. Schütz, A. de La Peña, R. Streblow, and D. Müller. Comparison between numerical models and experimental measurements of a cool down process in a thermal storage system. In *Energy for Sustainability 2015*, 2015.

[180] T. Schütz, M. H. Schraven, S. Remy, J. Granacher, D. Kemetmüller, M. Fuchs, and D. Müller. Optimal design of energy conversion units for residential buildings considering German market conditions. *Energy*, 139:895–915, 2017. doi:10.1016/j.energy.2017.08.024.

[181] A. M. Schweidtmann and A. Mitsos. Deterministic Global Optimization with Artificial Neural Networks Embedded. *Journal of Optimization Theory and Applications*, 180(3):925–948, 2019. doi:10.1007/s10957-018-1396-0.

[182] A. M. Schweidtmann, D. Bongartz, D. Grothe, T. Kerkenhoff, X. Lin, J. Najman, and A. Mitsos. Deterministic global optimization with Gaussian processes embedded. *Mathematical Programming Computation*, 13(3):553–581, 2021. doi:10.1007/s12532-021-00204-y.

[183] K. Seo, A. P. Retnanto, J. L. Martorell, T. F. Edgar, M. A. Stadtherr, and M. Baldea. Simultaneous design and operational optimization for flexible carbon capture process using ionic liquids. *Computers & Chemical Engineering*, 178:108344, 2023. doi:10.1016/j.compchemeng.2023.108344.

[184] U. Setzmann and W. Wagner. A New Equation of State and Tables of Thermodynamic Properties for Methane Covering the Range from the Melting Line to 625 K at Pressures up to 100 MPa. *Journal of Physical and Chemical Reference Data*, 20 (6):1061–1155, 1991. doi:10.1063/1.555898.

[185] O. Shcherbina, A. Neumaier, D. Sam-Haroud, X.-H. Vu, and T.-V. Nguyen. Benchmarking Global Optimization and Constraint Satisfaction Codes. In C. Bliek, C. Jermann, and A. Neumaier, editors, *Global Optimization and Constraint Satisfaction*, pages 211–222, Berlin, Heidelberg, 2003. Springer. doi:10.1007/978-3-540-39901-8_16.

[186] Y. Shi, J. Wang, and Y. Zhang. Sliding mode predictive control of main steam pressure in coal-fired power plant boiler. *Chinese Journal of Chemical Engineering*, 20(6):1107–1112, 2012. doi:10.1016/S1004-9541(12)60594-1.

[187] A. B. Singer and P. I. Barton. Global Optimization with Nonlinear Ordinary Differential Equations. *Journal of Global Optimization*, 34(2):159–190, 2006. doi:10.1007/s10898-005-7074-4.

[188] A. B. Singer, J. W. Taylor, P. I. Barton, and W. H. Green. Global Dynamic Optimization for Parameter Estimation In Chemical Kinetics. *Journal of Physical Chemistry A*, 110(3):971–976, 2006. doi:10.1021/jp0548873.

[189] H. Sirisena. Computation of optimal controls using a piecewise polynomial parameterization. *IEEE Transactions on Automatic Control*, 18(4):409–411, 1973. doi:10.1109/TAC.1973.1100329.

[190] E. M. Smith and C. C. Pantelides. Global optimisation of nonconvex MINLPs. *Computers & Chemical Engineering*, 21:791–796, 1997. doi:10.1016/S0098-1354(97)87599-0.

[191] E. Smith and C. C. Pantelides. A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. *Computers & Chemical Engineering*, 23(4–5):457–478, 1999. doi:10.1016/S0098-1354(98)00286-5.

[192] D. Steen, M. Stadler, G. Cardoso, M. Groissböck, N. DeForest, and C. Marnay. Modeling of thermal storage systems in MILP distributed energy resource models. *Applied Energy*, 137:782–792, 2015. doi:10.1016/j.apenergy.2014.07.036.

[193] C. Steffens, M. Pesavento, and M. E. Pfetsch. A Compact Formulation for the $l_{2,1}$ Mixed-Norm Minimization Problem. *IEEE Transactions on Signal Processing*, 66 (6):1483–1497, 2018. doi:10.1109/TSP.2017.2788431.

[194] E. W. Steyerberg, M. J. Eijkemans, F. E. Harrell, and J. D. F. Habbema. Prognostic modelling with logistic regression analysis: a comparison of selection and estimation methods in small data sets. *Statistics in Medicine*, 19(8):1059–1079, 2000. doi:10.1002/(SICI)1097-0258(20000430)19:8<1059::AID-SIM412>3.0.CO;2-0.

[195] M. Tawarmalani and N. V. Sahinidis. *Convexification and global optimization in continuous and mixed-integer nonlinear programming: Theory, Algorithms, Software, and Applications*, volume 65. Springer, New York, NY, 2002. doi:10.1007/978-1-4757-3532-1.

[196] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2):225–249, 2005. doi:10.1007/s10107-005-0581-8.

[197] J. W. Taylor, G. Ehlker, H.-H. Carstensen, L. Ruslen, R. W. Field, and W. H. Green. Direct Measurement of the Fast, Reversible Addition of Oxygen to Cyclohexadienyl Radicals in Nonpolar Solvents. *The Journal of Physical Chemistry A*, 108(35):7193–7203, 2004. doi:10.1021/jp0379547.

[198] N. Thissen, J. Hoffmann, S. Tigges, D. A. M. Vogel, J. J. Thoede, S. Khan, N. Schmitt, S. Heumann, B. J. M. Etzold, and A. K. Mechler. Industrially Relevant Conditions in Lab–Scale Analysis for Alkaline Water Electrolysis. *ChemElectroChem*, 11(1):e202300432, 2024. doi:10.1002/celc.202300432.

[199] N. Thissen, S. Khan, and A. K. Mechler. Electrochemical data on alkaline water electrolysis, 2024. doi:10.5281/zenodo.11103701.

[200] A. Tsoukalas and A. Mitsos. Multivariate McCormick relaxations. *Journal of Global Optimization*, 59(2):633–662, 2014. doi:10.1007/s10898-014-0176-0.

[201] V.-A. Vallianou and C. A. Frangopoulos. Dynamic Operation Optimization of a Trigeneration System. *International Journal of Thermodynamics*, 15(4), 2012. doi:10.5541/ijot.382.

[202] R. Vanderbei and colleagues. Nonlinear Optimization Models, 2023. URL `https://vanderbei.princeton.edu/ampl/nlmodels/`, Last accessed: Dec 1, 2024.

[203] S. Vigerske and A. Gleixner. SCIP: Global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. *Optimization Methods and Software*, 33 (3):563–593, 2018. doi:10.1080/10556788.2017.1335312.

[204] A. F. Villaverde, D. Henriques, K. Smallbone, S. Bongard, J. Schmid, D. Cicin-Sain, A. Crombach, J. Saez-Rodriguez, K. Mauch, E. Balsa-Canto, P. Mendes, J. Jaeger, and J. R. Banga. BioPreDyn-bench: a suite of benchmark problems for dynamic modelling in systems biology. *BMC Systems Biology*, 9:8, 2015. doi:10.1186/s12918-015-0144-4.

[205] A. F. Villaverde, F. Fröhlich, D. Weindl, J. Hasenauer, and J. R. Banga. Benchmarking optimization methods for parameter estimation in large kinetic models, 2018. doi:10.5281/zenodo.1304034.

[206] A. F. Villaverde, F. Fröhlich, D. Weindl, J. Hasenauer, and J. R. Banga. Benchmarking optimization methods for parameter estimation in large kinetic models. *Bioinformatics (Oxford, England)*, 35(5):830–838, 2019. doi:10.1093/bioinformatics/bty736.

[207] P. Voll. *Automated optimization based synthesis of distributed energy supply systems.* Dissertation, RWTH Aachen University, Aachen, 2014.

[208] P. Voll, S. Kirschbaum, and A. Bardow. Evaluation of Quasi-Stationary Simulation for the Analysis of Industrial Energy Systems. In *ecos 2010 : 23rd International Conference on Efficiency, Cost, Optimization, Simulation & Environmental Impact of Energy System, June 14-20, 2010, Lausanne, Switzerland ; book of papers*, pages 34–41, 2010.

[209] P. Voll, C. Klaffke, M. Hennen, and A. Bardow. Automated superstructure-based synthesis and optimization of distributed energy supply systems. *Energy*, 50:374–388, 2013. doi:10.1016/j.energy.2012.10.045.

[210] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006. doi:10.1007/s10107-004-0559-y.

[211] H. M. Wadsworth, editor. *Handbook of statistical methods for engineers and scientists*. McGraw-Hill, New York, 1 edition, 1990. ISBN 0070676747.

[212] W. Wagner and A. Pruß. The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use. *Journal of Physical and Chemical Reference Data*, 31(2):387–535, 2002. doi:10.1063/1.1461829.

[213] C. Wang, Q. Liu, and B. Qu. Global saddle points of nonlinear augmented Lagrangian functions. *Journal of Global Optimization*, 68(1):125–146, 2017. doi:10.1007/s10898-016-0456-y.

[214] J. Wei and M. J. Realff. Sample average approximation methods for stochastic MINLPs. *Computers & Chemical Engineering*, 28(3):333–346, 2004. doi:10.1016/S0098-1354(03)00194-7.

[215] A. K. Wilkins, B. Tidor, J. White, and P. I. Barton. Sensitivity Analysis for Oscillating Dynamical Systems. *SIAM Journal on Scientific Computing: A publication of the Society for Industrial and Applied Mathematics*, 31(4):2706–2732, 2009. doi:10.1137/070707129.

[216] C. M. Williams, A. Ghobeity, A. J. Pak, and A. Mitsos. Simultaneous optimization of size and short-term operation for an RO plant. *Desalination*, 301:42–52, 2012. doi:10.1016/j.desal.2012.06.009.

[217] S. Wischhusen and G. Schmitz. Transient simulation as an economical analysis method for energy supply systems for buildings or industry. *Applied Thermal Engineering*, 24(14):2157–2170, 2004. doi:10.1016/j.applthermaleng.2004.01.014.

[218] T. Yu and J. Q. Yuan. Modeling of the Coal-fired Utility Boiler in a Thermal Power Plant Based on Macro Energy Balance. In *Advanced Materials Research*, volume 732, pages 29–36, 2013. doi:10.4028/www.scientific.net/AMR.732-733.29.

[219] Q. Zhang, M. Martín, and I. E. Grossmann. Integrated design and operation of renewables-based fuels and power production networks. *Computers & Chemical Engineering*, 122:80–92, 2019. doi:10.1016/j.compchemeng.2018.06.018.

[220] Z. Zhou, P. Liu, Z. Li, E. N. Pistikopoulos, and M. C. Georgiadis. Impacts of equipment off-design characteristics on the optimal design and operation of combined cooling, heating and power systems. *Computers & Chemical Engineering*, 48:40–47, 2013. doi:10.1016/j.compchemeng.2012.08.007.