Global optimization in reduced space

by

Achim Wechsung

Dipl.-Ing., RWTH Aachen University (2008)

Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2014

Author .....
Department of Chemical Engineering
October 02, 2013

Certified by .....
Paul I. Barton
Lammot du Pont Professor of Chemical Engineering
Thesis Supervisor

Accepted by .....
Patrick S. Doyle
Professor of Chemical Engineering
Chairman, Committee for Graduate Students

Global optimization in reduced space
by
Achim Wechsung

Submitted to the Department of Chemical Engineering
on October 02, 2013, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Optimization is a key activity in any engineering discipline. Global optimization methods, in particular, strive to solve nonconvex problems, which often arise in chemical engineering, and deterministic algorithms such as branch-and-bound provide a certificate of optimality for the identified solution. Unfortunately, the worst-case runtime of these algorithms is exponential in the problem dimension. This leads to the notion of reduced-space problem formulations where either the number of variables that the algorithm branches on is reduced or only the actual degrees of freedom are visible to the optimization algorithms, following a partition of the variables into independent and dependent ones. This approach introduces new challenges though: McCormick relaxations, which are very easily applied in this setting, can be nonsmooth, the minima are very likely to be unconstrained causing the cluster problem and the information contained in the constraints is not as readily exploited.

In this thesis, several advances to both theory and methods are reported. First, a new analysis of the cluster problem is provided reaffirming the importance of second-order convergent bounding methods. The cluster problem refers to the phenomenon whereby a large number of boxes in the vicinity of a minimum are visited by branch-and-bound algorithms. In particular, it is shown that tighter relaxations can lead to a significant reduction in the number of boxes visited. Next, a constraint propagation technique for intervals is extended to McCormick relaxations. This reverse McCormick update utilizes information in the constraints and improves relaxations of the dependent variables, which can be used to either strengthen the relaxations of the feasible set or, using generalized McCormick relaxations, to construct reduced-space relaxations of the objective function. Third, a second-order convergent interval bounding method for the zeros of parametric nonlinear systems of equations is presented. This is useful to provide second-order convergent interval information to generalized McCormick relaxations, e.g., in the reverse propagation scheme. Fourth, the theory underpinning McCormick relaxations is extended to a class of discontinuous functions. It is further shown that branch-and-bound algorithms still possess their convergence properties.

Thesis Supervisor: Paul I. Barton
Title: Lammot du Pont Professor of Chemical Engineering

To my wife and my parents
for their love and support

Acknowledgments

> A diamond is a chunk of coal that is made good under pressure.
>
> *(Henry A. Kissinger)*

I am grateful for and appreciative of the support of many. First, I want to express my sincere gratitude towards Professor Paul Barton. I consider myself lucky to have had him as my thesis adviser. Paul was open to letting me shape my research instead of following a preconceived plan of his, available to discuss ideas for many hours when I needed guidance and advice, never shy to provide constructive, honest feedback, from which my work certainly benefited, and interested in educating the "whole individual"—as he likes to put it. Overall, his attitude also positively influenced the atmosphere in the group.

Next, I want to recognize my thesis committee, Professors William Green and George Stephanopoulos, for their valuable input, discussions and feedback at various stages of the thesis project. Many thanks also to the Barton group members, past and present, for making the PSE lab such a great place to work in. I especially grew fond of each and everyone's willingness to discuss ideas at any time, to provide helpful advice and to help out with technical and non-technical problems. I want to specially thank Joe Scott, Matt Stuber, Spencer Schaber, Kamil Khan, Kai Höffner and Stuart Harwood, who helped me at some stage of the thesis.

I want to thank my friends, whether at MIT, in Cambridge or at home, especially, Mike and Christy, for two fun years at 36 Adrian; Spencer, for being a great companion in the lab, on the road and on the water; Rachel, for organizing many great social activities in and beyond Cambridge; Drew and Stacey, for many fun dinners in the neighborhood; Daniel, for keeping our friendship from back-then alive across such large distance.

I owe much debt to my family who certainly shaped me more than anyone else. I am immensely grateful to my parents for providing a loving, nurturing and care-free home, for being incredibly supportive throughout my educational career and for letting me explore the world; to my siblings for great companionship while I was growing up and continuing now, as we are all adults, and for getting me away from my books from time-to-time; my grand-parents, none of whom lived to see me graduate and each of whom I remember dearly; and my parents-/siblings-in-law, for welcoming me into their homes and lives.

Finally, I cannot put into words how deeply grateful I am to my wife, the love of my life. If it were not for Sheetal, I would not have dreamed of going to, let alone graduating from, MIT. She is my greatest supporter and my source of strength. I admire her ability to put our relationship second so that I could take advantage of this opportunity and I am thankful for all the understanding, support and love she gave me even as she was busy working on her own thesis.

Lastly, I want to acknowledge Statoil and Siemens for supporting this research.

Contents

1 Introduction **19**
- 1.1 Branch-and-bound methods . . . 21
  - 1.1.1 Bounding methods . . . 23
  - 1.1.2 Domain reduction methods . . . 23
- 1.2 Full-space and reduced-space problem formulation . . . 25
  - 1.2.1 Regularity of the reduced-space model . . . 27
  - 1.2.2 Cluster effect for problems in the reduced-space formulation . . . 27
- 1.3 Contributions . . . 28

2 The cluster problem in global optimization **29**
- 2.1 Analysis of the cluster problem . . . 31
  - 2.1.1 Refinement of Neumaier's argument for a bound on the number of boxes necessary to cover B . . . 33
  - 2.1.2 A new analysis of the cluster problem . . . 34
- 2.2 Discussion of Theorem 2.1 . . . 37
- 2.3 Cluster problem for problems with non-differentiable functions . . . 38
- 2.4 Conclusion . . . 41

3 Factorable functions and methods to bound their range **43**
- 3.1 Concept of factorable functions . . . 43
  - 3.1.1 Basic definition . . . 43
  - 3.1.2 Representation as directed acyclic graph . . . 45
- 3.2 Interval analysis . . . 45
  - 3.2.1 Natural interval extensions . . . 48
  - 3.2.2 Centered forms . . . 49
- 3.3 McCormick analysis . . . 51
  - 3.3.1 Natural McCormick extensions . . . 54
  - 3.3.2 Standard McCormick relaxations . . . 55
- 3.4 αBB relaxations . . . 56
- 3.5 Comparison of bounding methods . . . 57

4 Reverse propagation of McCormick relaxations **59**
- 4.1 Reverse interval propagation . . . 60
  - 4.1.1 Reverse interval updates of binary operations . . . 62
  - 4.1.2 Reverse interval updates of univariate functions . . . 63

4.2 Reverse McCormick propagation . . . 64
4.2.1 Reverse McCormick updates of binary operations . . . 66
4.2.2 Reverse McCormick updates of univariate functions . . . 67
4.2.3 Inclusion monotonicity of the reverse McCormick updates . . . 68
4.2.4 Coherent concavity of the reverse McCormick updates . . . 70
4.3 Using reverse McCormick propagation in CSPs and in global optimization 72
4.3.1 Solving CSPs with equality and inequality constraints . . . 72
4.3.2 Constructing relaxations for reduced-space optimization problems . 75
4.3.3 Partitioning variables . . . 77
4.4 Implementation . . . 77
4.5 Case studies . . . 78
4.5.1 Equality constraints . . . 79
4.5.2 Inequality constraints . . . 81
4.5.3 Objective function . . . 83
4.6 Conclusion . . . 83

5 Second-order interval bounds for implicit functions 87
5.1 Preliminaries . . . 88
5.1.1 Relevant definitions and results from interval analysis . . . 89
5.2 Convergence order of parametric interval Newton methods . . . 91
5.3 Sensitivity-based bounding method . . . 95
5.3.1 Obtaining an initial enclosure of the sensitivities . . . 96
5.3.2 Improving the sensitivity bound . . . 98
5.3.3 Second-order convergent bounding method . . . 99
5.4 Case studies . . . 102
5.5 Conclusion . . . 105

6 Global optimization of discontinuous functions 107
6.1 Relaxations of bounded $\mathcal{L}$-factorable functions . . . 109
6.1.1 Extension of McCormick's result to bounded $\mathcal{L}$-factorable functions 109
6.1.2 Univariate piecewise continuous functions . . . 111
6.1.3 Examples of constructed relaxations . . . 112
6.1.4 Assumptions on f, o_k and the interval and McCormick extension of o_k 113
6.1.5 Discussion of sufficient conditions for convergence of the relaxations 116
6.1.6 Relaxations on sequences of intervals . . . 120
6.2 Branch-and-bound for bounded factorable optimization . . . 123
6.2.1 Convergence results when minimum is attained . . . 125
6.2.2 More general convergence results for branch-and-bound algorithm 127
6.3 Case Studies . . . 129
6.3.1 Process design and equipment sizing . . . 131
6.3.2 Discrete-time hybrid systems . . . 135
6.4 Conclusion . . . 137

7 Improving convergence of relaxations of bounded $\mathcal{L}$-factorable functions 139
7.1 Branching on discontinuous factors . . . 139
7.1.1 Validity of obtained lower bounds . . . 140
7.1.2 Consistency of bounding operation . . . 142
7.1.3 Certainty in the limit of the deletion by infeasibility rule . . . 144
7.2 Implementation details . . . 145
7.3 Case studies . . . 146
7.3.1 Motivating example revisited . . . 147
7.3.2 Parameter estimation with embedded dynamic model . . . 148
7.4 Discussion . . . 150
7.5 Conclusion . . . 153

8 Conclusion 155
8.1 Future work . . . 156

A Domain reduction using subgradients 159

B Synthesis of heat exchanger networks at subambient conditions 161
B.1 Introduction . . . 161
B.2 Problem statement . . . 163
B.3 Description of the process model . . . 164
B.3.1 A state space approach for design of heat exchanger networks including compressors and expanders . . . 164
B.3.2 A PA approach for the structure of the HEN and C&E system . . . 166
B.4 Model formulation . . . 173
B.4.1 The Pinch Operator . . . 173
B.4.2 The Pressure Operator . . . 176
B.4.3 The Exergy Operator . . . 177
B.4.4 The Objective Function . . . 178
B.5 Examples . . . 178
B.5.1 A simple example . . . 179
B.5.2 Design of an LNG process using LCO2 and LIN as cold carriers . . . 183
B.6 Discussion . . . 191
B.7 Conclusions . . . 193

C Pinch operator for streams with non-constant heat capacity 195
C.1 Utility targeting for streams with non-constant heat capacity . . . 195
C.1.1 Reformulating the targeting problem . . . 198
C.2 Heat exchanger network synthesis for streams with non-constant heat capacity 200

List of Figures

1.1 In a typical problem, the branch-and-bound algorithm identifies the solution quickly and spends most time improving the lower bound. 20

2.1 When the cluster problem occurs, a very large number of nodes is visited in the immediate vicinity of global solutions and near-optimal local solutions as shown in (a) and (b). An improved bounding method can mitigate this phenomenon effectively, see (c) and (d). 30

2.2 Illustration of different cases for a circle where dashed regions show boxes required to cover $\tilde{B}$ 36

2.3 Illustration of different cases for an ellipse where dashed regions show boxes required to cover $\tilde{B}$ 37

3.1 Directed acyclic graph of the $\mathcal{L}$-computational sequence given in Table 3.1 for the $\mathcal{L}$-factorable function $\mathbf{f}$ in Example 3.1 46

4.1 Illustration of domain reduction by reverse interval and McCormick propagation. The gray area is the set of all feasible solutions, the dash-dotted line is the original domain, the dotted line is the reduced domain using reverse interval propagation. The solid and dashed lines are relaxations of the feasible region parametrized by $\mathbf{p}$. 60

4.2 Principle of forward-reverse McCormick update to construct relaxations of the implicit set-valued mapping $\mathbf{x}(\cdot)$: forward evaluation of relaxation functions [156] to obtain a particular kind of relaxations of $\mathbf{g}$ and $\mathbf{h}$ on P (1), intersection with constraint information (2), and reverse propagation of additional information (3). This procedure can be iterated on if desired (4). 74

4.3 Result of reverse McCormick propagation for Example 4.1 showing the original bounds (dashed-dotted lines), the improved bounds (gray box), the convex and concave relaxations (solid red line, respectively) as well as results of the set-valued mapping $x(p)$ (asterisks). In (a) one iteration of the reverse McCormick propagation was performed while in (b) the reverse propagation iterations was repeated ten times. The dashed blue lines show convex and concave relaxations calculated using one iteration of the more expensive method in [164]. 79

4.4 Result of reverse McCormick propagation for Example 4.2 showing the original bounds (dashed-dotted lines), the improved bounds (gray box), the convex and concave relaxations (solid red lines) as well as results of the set-valued mapping $x(p)$ (asterisks). Additionally, zero level sets of the McCormick relaxations of $h(z, p)$ constructed on $X \times P$ (short dashed green lines) as well as $\tilde{X} \times \tilde{P}$ (dashed blue lines) are shown except where they are outside the interval bounds. Here, the results for different $P \times X$ are shown in (a) and (b). 80
4.5 Result of reverse McCormick propagation for Example 4.3 showing the original bounds (dashed-dotted lines), the improved bounds (gray box), the convex and concave relaxations (solid red lines) as well as results of the set-valued mapping $x(p)$ (asterisks). 81
4.6 Result of reverse McCormick propagation for Example 4.4 showing the original bounds (dashed-dotted lines), the improved bounds (gray box), the convex and concave relaxations (solid red lines) as well as results of the set-valued mapping $x(p)$ (asterisks). While in (a) the original bounds are used, in (b) the result of the bounds obtained from reverse interval propagation is shown. 82
4.7 Result of reverse McCormick propagation for Example 4.5 showing the original bounds (dotted lines), the improved bounds (dashed lines), the convex and concave relaxations (solid lines) as well as results of the set-valued mapping $x(p)$ (asterisks). 82
4.8 Result of reverse McCormick propagation for Example 4.6 showing the original bounds (dotted lines), the improved bounds (gray box), the convex and concave relaxations (solid red lines) as well as results of the set-valued mapping $x(p)$ (asterisks). 83
4.9 Result of reverse McCormick propagation for Example 4.7. In (a) the original bounds (dashed-dotted line), the improved bounds (gray box), the objective function f (asterisks) and the convex relaxations (red line) are shown as well as standard convex McCormick relaxations constructed on Y (green short dashed line) and $\tilde{Y}$ (blue dashed line) in a section. In (b) f is shown as a mesh and relaxations are shown as surfaces. 84

5.1 Empirical convergence order of the parametric Hansen-Sengupta operator (red line) and the parametric Krawczyk operator (green asterisks) for Example 5.1 93
5.2 Empirical convergence order of the sensitivity based bounding method for Example 5.2 103
5.3 Empirical convergence order of the sensitivity based bounding method for Example 5.3 103
5.4 Empirical convergence order of the sensitivity based bounding method (blue line), the Hansen-Sengupta operator (red line) and the Krawczyk operator (green asterisks) for Example 5.4 104

5.5 Empirical convergence order of the sensitivity based bounding method (blue line), the Hansen-Sengupta operator (red line) and the Krawczyk operator (green asterisks) for Example 5.5 105

6.1 Graph of f_1 (indicated by $+$) as well as convex relaxations (dashed line) and concave relaxations (continuous line) constructed on several intervals. Note that the scales on the vertical axes differ. 113
6.2 Graph of f_2 as well as its convex and concave relaxations on $[0.5, 1.5]^2$. Note that the scales on the vertical axes differ. 114
6.3 Graph of f (indicated by $+$) as well as five of its convex and concave relaxations (indicated by dashed and continuous lines, respectively) for $l = 1, 2, 4, 8, 16$. 116
6.4 Illustrations for Assumption 6.2 when $X \subset \mathbb{R}^2$. The curves indicate discontinuities introduced at previous factors. 118
6.5 Structure of heat exchanger network 1 132
6.6 Structure of heat exchanger network 2 134

7.1 Comparing the convergence behaviour of the upper and lower bounds without and with branching on discontinuous factors for the parameter estimation problem 149
7.2 Objective function of parameter estimation problem with embedded discrete-time hybrid system 150
7.3 Convergence behavior of lower bounds where interval and relaxations are used to compute bounds 150

B.1 Illustration of the natural sequence in process design (Reactor system, Separation system, Compression and Expansion, Heat exchanger system, Utilities) 162
B.2 State space realization of a heat exchanger and compressor/expander network including the exergy operator that transforms energies into exergies to quantify irreversibilities 165
B.3 Superstructure with heat exchangers, compressors and expanders for a hot and a cold stream split into segments showing intermediate temperatures . 167
B.4 Composite Curves resulting from compression of a hot stream at varying compressor intake temperatures 170
B.5 Exergy efficiency, required work and utilities for the example 172
B.6 Possible arrangement of streams in the simple example 179
B.7 Composite and Grand Composite Curves for the different cases in the simple example 182
B.8 The Liquefied Energy Chain 183
B.9 Process flow diagram of the base case offshore LNG process before pressure manipulation 185
B.10 Composite curves for the offshore LNG process before pressure manipulation and after application of the ExPAnD methodology 185

B.11 Process flow diagram of the offshore LNG process after applying the Ex-PAnD methodology 186
B.12 Final process flow diagram for the offshore LNG process 187
B.13 Compositive curve for the offshore LNG process resulting from the different optimization cases 190

C.1 Illustration of the basic concept of pinch analysis in the case of constant heat capacity flowrates 196
C.2 Illustration of the second law constraint for feasibility of the heat exchange for process streams with nonconstant heat capacity 198

List of Tables

2.1 The cluster problem is very sensitive to the termination tolerance. The employed bounding methods correspond to those used to construct Figure 2.1. 29
2.2 Summary of results for number of boxes required to cover $\tilde{B}$ when $\beta = 2$. 38

3.1 One possible representation of $\mathbf{f}$ in Example 3.1 as a $\mathcal{L}$-computational sequence. 45
3.2 Comparison of the computational complexity and the convergence order of different bounding methods. For reference, the complexity of one evaluation of the natural function is $O(n_f)$. 57

6.1 Factorization of $f = \psi(x) - \psi(x)$ on $X^l = [-l^{-1}, l^{-1}]$. 115
6.2 Equipment cost correlation for heat exchangers depending on required area 132
6.3 Data for process and utility streams in heat exchanger network 1 133
6.4 Comparison of different methods with BARON for the first heat exchanger case study 133
6.5 Data for process and utility streams in heat exchanger network 2 134
6.6 Comparison of different methods with BARON for the second heat exchanger case study 135
6.7 Comparison of different methods for both cases of the discrete-time hybrid system. Note that Method 1 does not converge in either case after solving 100,000 iterations. 136

7.1 Nodes visited and bounds calculated for the motivating example 147
7.2 Experimentally measured concentration of product in the reactor effluent . 149
7.3 Comparison of different methods for the parameter estimation problem with embedded discrete-time hybrid system. 151
7.4 Influence of heuristic for discontinuous branching on convergence of second case study using Method 3 152

B.1 Effect of compression of a hot stream at varying compressor intake temperatures on utility requirements and exergy efficiency 169
B.2 Given information for stream in simple example 179
B.3 Result for decision variables for Case 2 of the simple example 180
B.4 Result for decision variables for Case 3 of the simple example 181
B.5 Result for decision variables for Case 4 of the simple example 181

B.6 Main results for LNG case study. I refers to the base case design, II after application of the ExPAnD methodology, IIIa–d refer to the different optimization scenarios. $W > 0$ indicates that work needs to be supplied while $W < 0$ means that work is generated. 184
B.7 Given data for the optimization of the offshore LNG process 188
B.8 Results for the decision variables for Case III of the LNG offshore process design 189

Chapter 1

Introduction

Optimization [29, 30, 130] is a key activity in any engineering discipline. Given a mathematical description of the problem at hand and a metric by which various alternatives can be ranked, different existing technical solutions to a problem can be compared and new proposed solutions can be easily evaluated and accepted if better or discarded if not. Optimization in the strictest sense does not just imply improvement of a previous solution; it refers to finding the truly best solution(s) to a problem. When the optimization problem is convex, any local solution is also a global solution [35]. Furthermore, efficient algorithms for this class of problems are known [82, 125]. Nonconvex problems, on the other hand, may possess suboptimal local solutions. No algorithm whose runtime is polynomial in the inputs is known for this problem class[1]. Methods for this task can be assigned to different categories depending on their run-time behavior [129]. In this thesis, we consider complete deterministic methods for nonconvex optimization only. *Complete* methods can give an approximate solution within a specified tolerance in finite time. *Deterministic* means that the behavior of the algorithm depends solely on its data and does not change from execution to execution.

Complete methods for global optimization provide conservative estimates of how much a current solution could still be improved, and are not prone to find solutions that are locally optimal only. While good initial guesses certainly improve their run-time, they do not rely on these. On the contrary, their ability to identify the global solution is independent of the initial user-specified point [88].

In order to provide a guarantee that a (nearly) global solution has been found, complete global optimization methods require a means to bound the objective function conservatively on subsets of its domain. Experience shows that most computational effort is directed towards successively refining, or improving, this bound; cf. Figure 1.1. This need for global information about the problem is the single most important distinction with local methods for optimization [130]. Simplifying, one can say that local methods process local information such as function values or gradients at the current iterate only. This data is used to calculate the next iterate. At each iteration of a global method, on the other hand, information about the complete domain, or at least the currently considered subset thereof, is needed. Obviously, obtaining accurate global information is as hard as solving the original problem, so tractable conservative procedures have been designed for this task, some of which are studied in detail in this thesis.

[1] In fact, it is conjectured that such algorithms do not exist.

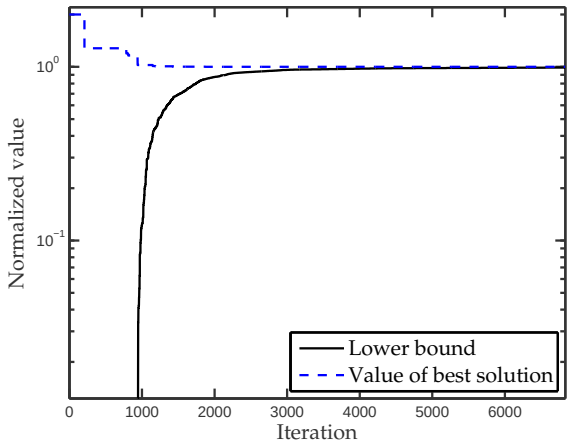


Figure 1.1: In a typical problem, the branch-and-bound algorithm identifies the solution quickly and spends most time improving the lower bound.

While in some engineering disciplines linear models are sufficiently accurate to represent the physical phenomena, chemical process systems often require nonlinear models. For example, even simple networks of mixers and splitters with multi-component flows result in nonlinear and nonconvex models [78]. Typical process flowsheet models of chemical plants are significantly more complex. While local optimization methods have been successfully applied to such problems, the quality of the identified solution depends strongly on the initial guess which, in turn, depends on the user's understanding of the process. Commercially available steady-state process simulators such as Aspen Plus® or HYSYS® allow for the construction of complicated flowsheets with complex physical process models. In these software packages, "rigorous" blocks with tailored algorithms for each unit operation are connected to build a model of the process [33]. Embedded local optimization routines can be used to improve an initial guess. Work on building process simulators that allow for global optimization is still in its infancy [11, 36, 37].

The most general problem can be described as follows: suppose $D \subset \mathbb{R}^n$ is open, $C \subset D$ is convex and let $f : D \to \mathbb{R}$, $\mathbf{g} : D \to \mathbb{R}^{n_g}$ and $\mathbf{h} : D \to \mathbb{R}^{n_h}$. Consider the optimization problem given by

$$\begin{aligned} f^* = \inf_{\mathbf{y} \in C} \quad & f(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{y}) \leq \mathbf{0}, \\ & \mathbf{h}(\mathbf{y}) = \mathbf{0}. \end{aligned} \tag{1.1}$$

The *feasible set* of (1.1) is given by $E = \{\mathbf{y} \in C : \mathbf{g}(\mathbf{y}) \leq \mathbf{0}, \mathbf{h}(\mathbf{y}) = \mathbf{0}\}$. Any $\mathbf{y}^*$ in the feasible set that satisfies $f(\mathbf{y}^*) = f^*$ is called a *minimum* or a *solution*, sometimes with the qualifier *optimal*, of (1.1). f^* is known as the *infimum* or the *optimal solution value*. When $f^* = -\infty$, (1.1) is said to be *unbounded*. If the feasible set is empty, then (1.1) is said to be *infeasible*.

In practical applications, $\mathbf{h}$ is often obtained from mass, species, momentum or energy balances, thermodynamic property models, kinetic rate expressions or similar fundamental physical or chemical laws or correlations, whereas $\mathbf{g}$ typically describes either technical process limitations or desired product qualities and f is usually an economic or performance metric, for example, the net present value, profit or energy efficiency.

In order to guarantee properties such as convergence of the numerical algorithms used to find $\mathbf{y}^*$ and f^*, certain regularity assumptions are typically required to hold for C, f, $\mathbf{g}$ and $\mathbf{h}$.

1.1 Branch-and-bound methods

Many deterministic global optimization methods use a continuous branch-and-bound algorithm at their core [59, 88]. A branch-and-bound algorithm in its basic form is given as Algorithm 1.1. Here, we associate with each node a set, which can be easily characterized, e.g., an n-dimensional interval, also known as a box. Starting at the root node with a set that encloses all solutions to be considered, the algorithm repeatedly visits nodes, bounds the optimal solution on the current node and, if improvement is possible, looks for a better feasible solution, otherwise, the box is discarded. If a solution better than previously known has been found, it is stored. The current box is partitioned into two smaller boxes, which are stored along with the lower bound on the parent box for future processing. A new overall lower bound on the objective value is established as the minimum of all lower bounds of the remaining stored boxes. All boxes where the lower bound is greater than the best identified solution value can be safely discarded. Until the best identified solution value and the current lower bound agree to within some pre-specified tolerance, the procedure is repeated.

Even in most basic form, Algorithm 1.1 relies on two heuristics: the node selection heuristic [88, p. 130] that decides which node to visit next and the branching heuristic [50, 140] that decides how to partition a node. General requirements for each, which are sufficient for finite convergence of the branch-and-bound algorithm, are given in [88].

This algorithm has exponential worst-case run-time dependence on the dimension of the problem. In other words, increasing the problem dimension by one, i.e., by introducing a single additional variable to the problem, can potentially double the time required to solve the problem globally. This phenomenon is often referred to as the "curse of dimensionality" [22]. Conceptually, adding new dimensions to the search space adds new degrees of freedom thus complicating the search for the best possible solution. Using so-called domain reduction techniques[2] is one avenue to mitigate this exponential behavior to some extent.

[2]Sometimes domain reduction methods are also referred to as range reduction or bounds tightening procedures.

Algorithm 1.1: Generic branch-and-bound algorithm

Input: box X_0, termination criterion
Output: problem infeasible ($f^* = +\infty$) or solution $\mathbf{x}^*$, solution value f^*

$LBD \leftarrow -\infty$, $f^* \leftarrow +\infty$, $k \leftarrow 1$, $\mathcal{N} \leftarrow \{X_0\}$;
while `NotConverged` *(LBD, f^*) and* $|\mathcal{N}| > 0$ **do**
  $X_k \leftarrow$`SelectBox` $(\mathcal{N})$;
  $\mathcal{N} \leftarrow \mathcal{N} \backslash \{X_k\}$;
  $LBD(X_k) \leftarrow$`LowerBound` (X_k);
  if $LBD(X_k) < f^*$ **then**
    $(UBD_k, \mathbf{x}_k) \leftarrow$`FindFeasibleSolution` (X_k);
    if $UBD_k < f^*$ **then**
      $f^* \leftarrow UBD_k$, $\mathbf{x}^* \leftarrow \mathbf{x}_k$;
      $\mathcal{N} \leftarrow \{X \in \mathcal{N} :$ `NotConverged`$(LBD(X), f^*)\}$;
    $X_k \leftarrow$`ReduceDomain` $(X_k, f^*, LBD(X_k))$; `// optional`
    $(X', X'') \leftarrow$`PartitionBox` (X_k);
    $LBD(X') \leftarrow LBD(X_k)$, $LBD(X'') \leftarrow LBD(X_k)$;
    $\mathcal{N} \leftarrow \mathcal{N} \cup \{X', X''\}$;
  $LBD \leftarrow \min_{X \in \mathcal{N}} LBD(X)$;
  $k \leftarrow k + 1$;
return $(f^*, \mathbf{x}^*)$;

1.1.1 Bounding methods

As outlined above, obtaining global information about the range of a function is a key component in a deterministic global optimization algorithm. We show in Chapter 2 that the features of the bounding procedure have a great influence on the number of iterations of a branch-and-bound algorithm. Summarizing, three properties are essential for an effective bounding method:

1. efficiency of the required computations,
2. initial overestimation by the method, and
3. rate of convergence of the resulting conservative bound to the true image.

While the first requirement is common to any numerical method, the remaining two can be loosely recapped as "tight on large boxes and rapidly converging".

Simplistically, one can distinguish two routes to bound the range of a function. Some methods directly compute a lower bound on the range of a given function whereas other methods set up a convex optimization problem that is guaranteed to return a lower bound on the range of the function.

The most important representatives belonging to the first class are various methods in the realm of interval analysis [3, 122, 127]. The basic object in interval analysis is the interval, to which the arithmetic in the real number system can be extended. This so-called interval arithmetic can then be used to estimate conservatively the range of a function. Methods that are derived in this context include natural interval extensions and centered forms. Other methods with similar ideas are also known [e.g., 161], but not further considered.

The latter class approaches the problem differently. Another optimization problem, termed a *relaxation* of (1.1), is derived from (1.1) by enlarging the feasible set and/or replacing the objective function with a function that takes a smaller value for each point in its domain. In addition, the relaxation is either a linear or a convex program so that it can be efficiently and reliably solved to global optimality [29, 30, 35]. Examples include McCormick's method [118, 156], αBB relaxations [1] and smooth reformulation with the introduction of auxiliary variables and constraints [159, 165, 166]. When the obtained relaxations are convex, it is in principle possible to linearize these and solve the resulting linear program [e.g., 167].

In Chapter 3, we study these different bounding methods in more detail.

1.1.2 Domain reduction methods

Domain reductions methods strive to shrink the currently considered box using information about the best solution found so far. Their goal is to discard a subset of the search space for which it can be established that either no feasible solution exists or that any feasible solution is no better than best solution found up to now. This step is optional in the sense that branch-and-bound algorithms are shown to converge without it, but it

can reduce the iteration count and the run time. Proposed methods include optimality-based [148], feasibility-based [75] and duality-based [166] techniques. It has also been suggested to construct conservative (outer approximating) linearizations of the constraint functions and minimizing or maximizing each variable as linear programs [36]. Other ideas are present in the literature under the keyword "constraint propagation" [e.g., 31].

A constraint satisfaction problem (CSP) consists of a finite set of variables, domains and constraints. A solution of a CSP is an assignment of values from the domains to the variables so that all constraints are satisfied. In general, these problems are NP-hard and hence it is desirable to compute an enclosure of the solution set. Constraint propagation routines, or, more generally, contractors, are numerical methods that assist in this task. Using the information about the relationship between variables that is contained in a single constraint, or in a set of constraints, they attempt to shrink the domains. Typically, intervals are used to enclose the solution sets whereas a constraint propagation technique for McCormick relaxations [118, 156] are proposed in this contribution.

Constraint propagation was first developed for logic constraints on discrete domains [114]. Different notions of consistency, which describes the degree to which the remaining elements of the domain satisfy the constraints, have been introduced for this case [13, 31]. Constraint propagation has also been applied to connected sets that appear in so-called numerical CSPs [28, 51] and a large number of techniques have been proposed in the literature.

Many constraint propagation methods use ideas from interval analysis: they consider interval domains and use interval arithmetic. Cleary [46] and Davis [51] presented the first algorithms for constraint propagation with interval domains. Hyvönen [89] considered cases where exact numbers are insufficient and studied how interval arithmetic can be utilized in CSPs. Lhomme [107] proposed an extension of arc-consistency to numeric CSPs. Benhamou et al. [26] introduced the notion of box-consistency. Sam-Haroud and Faltings [152] approximated feasible regions by 2^n-trees and presented algorithms to label leaves consistently. Benhamou and Older [25] proposed the notion of hull-consistency. Van Hentenryck et al. [169] showed how interval extensions can be used to calculate box-consistent labels, see also [170]. Benhamou et al. [27] proposed an algorithm for hull-consistency that does not require decomposing constraints into primitives. Vu et al. [173] proposed a method to construct inner and outer approximations of the feasible set using unions of intervals. Lebbah et al. [106] discussed how the reformulation-linearization technique can be used to relax nonlinear constraints and to aid in pruning the search space. Granvilliers and Benhamou [69] proposed an algorithm that prunes boxes using both constraint propagation techniques and the interval Newton method. Recently, Domes and Neumaier [53] proposed a constraint propagation method for linear and quadratic constraints and Jaulin [90] studied set-valued CSPs.

Jaulin et al. [91] discussed contractors based on interval analysis, many of which were also the subject of Neumaier's book, though it focused on solving systems of equations in the presence of data uncertainty [127]. Recently, Schichl and Neumaier [153] studied directed acyclic graphs (DAGs) to represent functions for interval evaluation. Vu et al. [174] used this representation and extended the contractor proposed in [27], which propagates

interval information forward and backward along the DAG. Recently, Stuber et al. [164] extended contractors based on interval analysis to compute convex and concave relaxations of implicit functions. However, their methods require existence and uniqueness of the implicit function on the full domain.

Continuous optimization problems are often solved to guaranteed global optimality using continuous branch-and-bound algorithms [59, 88]. It is well-known that the efficiency of these algorithms can be improved by discarding parts of the search space that are infeasible or that are known not to contain optimal solutions [165]. These tasks are often referred to as domain reduction. Obviously, global optimization is an important application of CSPs [129] and ideas originally developed for CSPs are routinely utilized in global optimization: logic-based methods can enhance and expedite optimization routines [86]; constraint propagation is often used to discard parts of the domain where the solution is known not to exist [e.g., 74, 75, 148]. For example, constraint propagation routines are part of BARON's pre-processing step [151]. It is also not coincidental that many constraint satisfaction tools use branch-and-prune frameworks inspired by global optimization algorithms to identify a set of boxes that contains all solutions [e.g., 69, 106, 169]. Also, see the recent discussion of feasibility-based bounds-tightening procedures in [23, 24]. Thus, borrowing and embracing ideas from the other field has been very beneficial for both fields.

As briefly described in Section 1.1.1, branch-and-bounds algorithms also require computable rigorous bounds on the objective function and on non-convex constraints. In Chapter 4, we explore how ideas from CSP can be used to construct improved relaxations of nonconvex functions, in particular, when these are defined implicitly only. Compared to the interval method in Vu et al. [174], the proposed method for McCormick relaxations goes one step further. It traverses the directed acyclic graph of a factorable function forwards and backwards. During the forward pass, the typical operations for McCormick relaxations [118, 156] are performed. The obtained relaxations are then tightened using information about the constraints and the graph is traversed in reverse order. At each node, the operation is inverted in a sense that is detailed in Section 4.2. In the end, we obtain tighter relaxations of each variable. Depending on the initialization of the relaxations of the independent variables prior to the forward pass, these can be interpreted in a different way as we discuss in detail in Section 4.3.

1.2 Full-space and reduced-space problem formulation

As outlined above, to each (1.1), there exists a directed acyclic graph[3]. One very common problem reformulation introduces additional variables and constraints for each intermediate node in the graph [159, 165–167]. In some sense, this results in an "increased-space" problem formulation. Main advantages of this approach include the fact that most constraints involve small numbers of variables only so that the reformulated problem is

[3]Typically, many different graphs exist that correspond to the same problem, which have relaxations of varying quality.

much more sparse and the ease with which a tight relaxation can be constructed for each nonconvex constraint of the reformulation. The disadvantage of the formulation is the large dimension of the lower bounding problem that necessitates a weaker linearization of the relaxations of the constraints to obtain a LP relaxation [167].

As mentioned earlier, all known global optimization algorithms scale exponentially with the problem dimension. Thus, it has been suggested [e.g., 56] to focus the attention of the optimizer on a select subset of the variables. For example, if it suffices to branch on a small subset of variables $\mathbf{y}$ in (1.1), then the curse of dimensionality can be potentially mitigated. In [56], a selective branching scheme is proposed and shown to converge for a class of functions. It is applicable when the objective function and each inequality constraint can be decomposed into the following form: $w(\mathbf{z}, \mathbf{p}) = w^A(\mathbf{z}) + \sum_i w^{B,i}(\mathbf{z}) w^{C,i}(\mathbf{p}) + w^D(\mathbf{p})$ where w^A and $w^{B,i}$ are convex and $w^{C,i}$ and w^D are continuous. Furthermore, for each i, $w^{B,i}$ must be affine or $w^{C,i}$ non-negative. Under these conditions, the authors show that it suffices to branch on $\mathbf{p}$ only. Note that it can only be applied to equality constraints when they are linear in $\mathbf{z}$.

Similarly, in [96] a pre-processing method is proposed to determine a minimum set of variables in the increased-space problem formulation that needs to be branched on in order to guarantee convergence. This paper extends earlier work in [94]. Other work to detect convexity of graphs include [63, 64].

However, the cost of each bounding problem that is solved in each iteration of the branch-and-bound algorithm also scales with the number of the variables. Thus, it might be beneficial if the problem could be recast so that only a subset of the variables is visible as far as every routine of the branch-and-bound algorithm is concerned.

Suppose that we can partition the variables into *independent* and *dependent* variables, $\mathbf{p} \in D_p \subset \mathbb{R}^{n_p}$ and $\mathbf{z} \in D_x \subset \mathbb{R}^{n_x}$, respectively, so that $n = n_p + n_x$ and we have $\mathbf{y} = (\mathbf{p}, \mathbf{z})$ for each $\mathbf{y} \in D$; similarly, we partition C into C_x and C_p. The intention is that the equations $\mathbf{h}(\mathbf{y}) = \mathbf{0}$ can be used to find exactly one $\mathbf{z} \in C_x$ for each $\mathbf{p} \in C_p$ so that $\mathbf{h}(\mathbf{p}, \mathbf{z}) = \mathbf{0}$. In other words, we use the equality constraint in (1.1) to define implicitly a mapping $\mathbf{x} : C_p \to C_x$. Consequently, we can transform (1.1) into

$$\begin{aligned} \min_{\mathbf{p} \in C_p} \quad & f(\mathbf{p}, \mathbf{x}(\mathbf{p})) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{p}, \mathbf{x}(\mathbf{p})) \leq \mathbf{0}, \end{aligned} \tag{1.2}$$

a problem with a smaller number of variables and less constraints. Thus, the optimizer sees only n_p instead of n variables. In a branch-and-bound algorithm, this also means that the lower and upper bounding problems are solved in this reduced space, decreasing the computational cost of each iteration which can be assumed to scale polynomially with the number of variables.

When we employ relaxations to construct the lower bounding problem, it is necessary to know relaxations of $\mathbf{x}$. Stuber et al. [164] presented a technique for the construction of such relaxations. As we briefly alluded to above, Chapter 4 also contains methods for obtaining relaxations of implicitly defined functions or set-valued mappings. These are

especially useful when applied to the reduced-space problem formulation as they allow us to construct relaxations of $\mathbf{x}$ on convex subsets of C_p.

1.2.1 Regularity of the reduced-space model

We also mentioned above that certain regularity assumptions are often required by the numerical methods employed. For example, gradient-based local search routines tacitly assume that the functions are at least continuously differentiable [e.g., 130]. When this assumption does not hold, their behavior is no longer guaranteed by most of the established theoretical results. As an illustrative example, consider the method of steepest descent, which uses the negative of the gradient vector to restrict the search to a ray originating at the current point. Along this ray, the best possible point is selected as the next iterate. For a differentiable function and a point in the interior of the feasible region, this is a sensible choice[4]: one can always find a better solution when moving in the direction of steepest descent. When the differentiability assumption is removed, this no longer holds. Similarly, when the functions are not continuous it is not even guaranteed that points "nearby" are also "close" in their objective function value.

However, in some problems, the modeled reality does indeed behave in such a non-regular fashion [e.g., 14]. In order to be able to use the standard numerical tools, models have been proposed that regularize the mathematical formulation by increasing the number of variables and/or constraints [e.g., 55, 168]. In Chapter 6, we present the first method to solve global optimization problems with discontinuities and compare our reduced-space approach to the increased-space formulation in the literature. However, the lack of regularity presents unique challenges; in particular, convergence of the employed relaxations can be slow or incomplete. Chapter 7 contains some improvements that can guarantee convergence of the relaxations.

1.2.2 Cluster effect for problems in the reduced-space formulation

Solving reduced-space problems makes it more likely that a solution is unconstrained: in the most extreme case ($n_g = 0$ and $n_h \gg 0$), the reduced space formulation lacks all of the constraints that were present in the full space formulation. Hence, it is much more likely that the optimal solution is in the interior of the feasible region of (1.2). As we will see in Chapter 2, unconstrained global optima of smooth optimization problems are known to be prone to the *cluster problem* [54], i.e., the branch-and-bound algorithm creates a large number of nodes in the immediate vicinity of the optimal solution. Since points close to the optimal solution are also close in objective function value in a second-order sense, and we require a verification of the global optimum within some tolerance, the algorithm can guarantee the optimality of the found solution only after solving the lower bounding problem on a large number of small boxes in the direct vicinity of the solution. We confirm the literature result that the convergence order of the bounding procedure greatly influences the severity of this effect. Also, we argue that the required convergence

[4] Although more efficient local optimization approaches are known; see [e.g., 130]

order criteria are easier to satisfy for non-smooth problems when the optimal solution is at a point of non-differentiability. Lastly, we add that constrained optimal solutions typically do not exhibit this behavior or at least not to the same extent [129]. This is due to the fact that most constrained minima are not also stationary points.

In Chapter 5, we also show that current parametric interval methods for nonlinear equations provide linearly convergent bounds on the set of zeros only. Based on the sensitivities, we present a second-order convergent parametric interval method. The obtained bounds can then be used to initialize generalized McCormick relaxations, e.g., for the reverse McCormick propagation as discussed in Chapter 4 in order to achieve the important quadratic convergence order of the relaxations. Also, it can serve as a more effective domain reduction method.

1.3 Contributions

This thesis contains these original contributions:

- a new analysis of cluster effect[5], Chapter 2,
- a method to improve McCormick relaxations using the information contained equality and inequality constraints, Chapter 4,
- a second-order convergent method for interval bounds of parametric nonlinear equations, Chapter 5,
- a method to solve discontinuous global optimization problems[6], Chapter 6,
- a method to improve convergence of relaxations of discontinuous functions, Chapter 7,
- a domain reduction technique based on subgradients of the convex relaxations of the objective function, Appendix A,
- a method for optimal process design with heat integration at subambient conditions[7], Appendix B,
- a pinch operator for streams with non-constant heat capacity, Appendix C.

[5]Published as [178]
[6]Published as [176]
[7]Published as [177]

Chapter 2

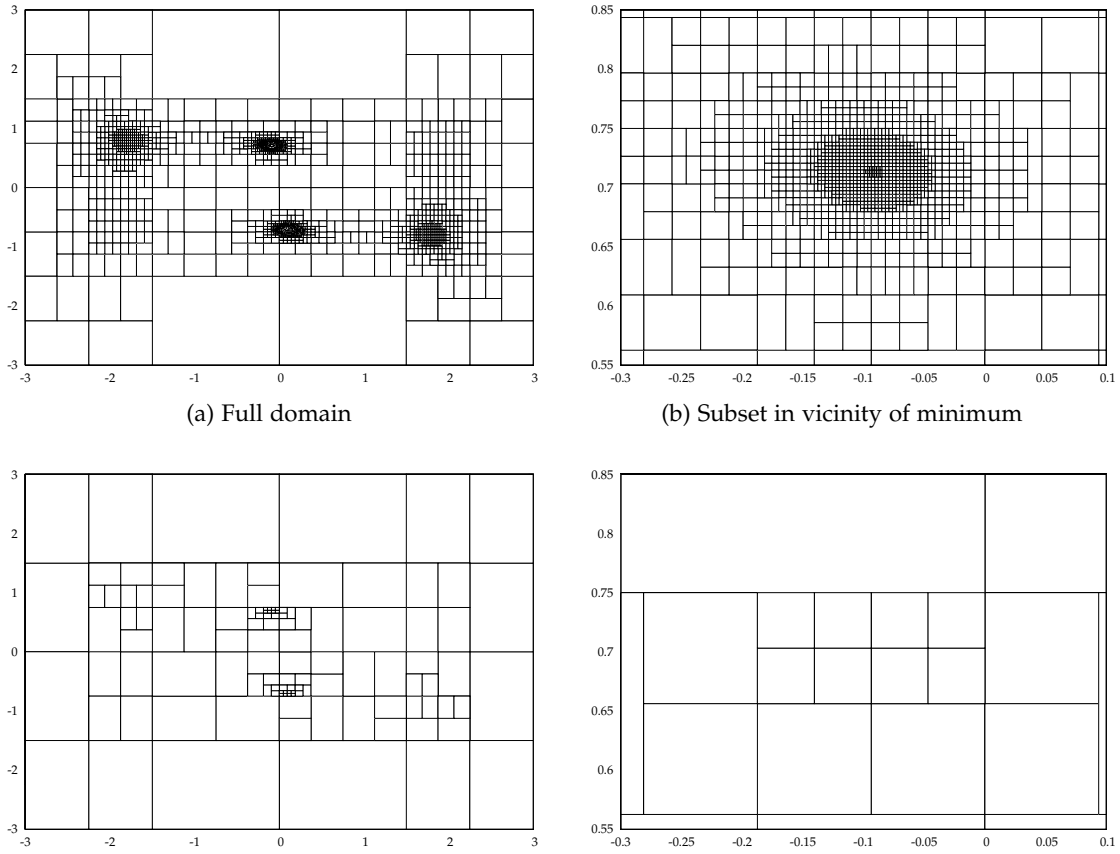
The cluster problem in global optimization

It is well known that branch-and-bound algorithms for continuous global optimization [59, 88] can create a large number of small boxes in the vicinity of a global minimizer, see Figure 2.1 and Table 2.1. This behavior was first discussed by Du and Kearfott [54] in the context of interval branch-and-bound methods and the authors coined the term *cluster problem* for this phenomenon. They provided an analysis to establish an upper bound on the number of boxes that cannot be fathomed by value dominance before the width of the boxes becomes smaller then a user specified tolerance. The authors were also the first to point out the importance of the convergence order of the bounding method (see Definition 2.1) in mitigating the cluster problem. Later, Neumaier [129] provided a similar analysis: it considers a hyperellipsoidal region around an unconstrained global minimizer, uses the determinant of the Hessian at the global minimizer instead of its smallest eigenvalue and introduces the proportionality constant for the volume of a hypersphere. Regardless, the result of the analysis is similar to Du and Kearfott [54] and stresses the importance of the convergence order. The main conclusion in these articles is that, in the worst case, at least second-order convergence is necessary to overcome the cluster problem. However, even with second-order convergence, the number of boxes still has exponential dependence on the problem dimension as Neumaier claims in [129]. Recently, Schöbel and Scholz [154] studied the worst-case behavior of branch-and-bound algorithms and gave an upper bound on the number of boxes needed for convergence that is very conservative.

If the minimizer coincides with the vertex of a box at some point in the branch-and-

Termination tolerance	Nodes visited by	
	Method 1	Method 2
0.1	2,091	171
0.01	6,831	231
0.001	56,531	275
0.0001	549,347	299
0.00001	> 1,000,000	355

Table 2.1: The cluster problem is very sensitive to the termination tolerance. The employed bounding methods correspond to those used to construct Figure 2.1.



(a) Full domain (b) Subset in vicinity of minimum

(c) Full domain (d) Subset in vicinity of minimum

Figure 2.1: When the cluster problem occurs, a very large number of nodes is visited in the immediate vicinity of global solutions and near-optimal local solutions as shown in (a) and (b). An improved bounding method can mitigate this phenomenon effectively, see (c) and (d).

bound algorithm then an exponential number of boxes will contain this minimizer. The analysis presented below assumes, however, that boxes can be placed so that the minimizer is in the center of the box. Strategies such as back-boxing [171] or epsilon-inflation [117] can potentially avoid the former case. Also, see the discussion in [129, Chapter 15].

Here, the cluster problem is revisited and the analysis is refined. In particular, it is shown that the convergence order pre-factor is important: assuming second-order convergence, the exponential dependence on the problem dimension can be avoided if the pre-factor is sufficiently small and the minimizer is always in the interior of a box in the branch-and-bound tree. Thus, not all relaxations with second-order convergence are equal in higher dimensions. On the contrary, tightness of the relaxations, for which the pre-factor is a good measure, is very important. Lastly, it is shown that for nonsmooth optimization problems where the objective function is not differentiable at the optimal solution, linear convergence of the relaxations can suffice to prevent the cluster problem.

2.1 Analysis of the cluster problem

It is assumed that the reader is familiar with branch-and-bound algorithms for continuous global optimization [59, 88], also see Section 1.1, and the construction and use of convex relaxations in such algorithms [2, 5, 118, 156], also see Chapter 3.

Assumption 2.1. Suppose $D \subset \mathbb{R}^n$ is open, $C \subset D$ is convex and let $f : D \to \mathbb{R}$ be twice differentiable on D. Suppose that $\mathbf{x}^*$ is the *unique unconstrained* global minimum of f on C, so that $\nabla f(\mathbf{x}^*) = \mathbf{0}$ and suppose furthermore that $\nabla^2 f(\mathbf{x}^*)$ is positive definite.

Suppose $Z \subset \mathbb{R}^n$. The *set of all interval subsets* of Z is denoted by $\mathbb{IZ}$. The *width* of an n-dimensional interval $X = [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$ is defined as $w(X) = \max_{i=1,\ldots,n}(\overline{x}_i - \underline{x}_i)$.

Definition 2.1. Let a continuous convex relaxation[1] of f on any $X \in \mathbb{IC}$ be given by $f_X : X \to \mathbb{R}$. The relaxations are said to have *convergence order* $\beta \geq 1$ if there exists $K > 0$ such that

$$\min_{\mathbf{x}\in X} f(\mathbf{x}) - \min_{\mathbf{x}\in X} \check{f}_X(\mathbf{x}) \leq K w(X)^\beta, \quad \forall X \in \mathbb{IC}. \tag{2.1}$$

Note that convergence order is typically defined as the difference of the width of the image of X under the enclosure of f and the width of the image of X under f [34, 128, 138]. However, Definition 2.1 is more natural for the purpose of this chapter and the difference is unimportant for this argument.

Assumption 2.2. Let ε be the termination tolerance for the branch-and-bound algorithm and assume the algorithm has found the upper bound, $UBD_k = f(\mathbf{x}^*)$. Assume the algorithm terminates at iteration k when $UBD_k - LBD_k \leq \varepsilon$, where LBD_k is the current lower bound.

[1] For a in-depth discussion refer to Chapter 3.

Lemma 2.1. *Let $X^* \in \mathbb{IC}$ be such that $\mathbf{x}^* \in X^*$. If the bound given by Definition 2.1 is sharp for all $X \in \mathbb{IC}$, then a necessary condition for termination of the branch-and-bound algorithm is*

$$w(X^*) \leq \left(\frac{\varepsilon}{K}\right)^{\frac{1}{\beta}}. \tag{2.2}$$

Proof. At any iteration $LBD_k \leq \min_{\mathbf{x}\in X^*} \check{f}_{X^*}(\mathbf{x}) \leq UBD_k$ holds. Thus, a necessary condition for termination is $UBD_k - \min_{\mathbf{x}\in X^*} \check{f}_{X^*}(\mathbf{x}) \leq \varepsilon$. In the worst case, the bound on the underestimation by the relaxation in (2.1) is exact so that

$$UBD_k - \min_{\mathbf{x}\in X^*} \check{f}_{X^*}(\mathbf{x}) = \min_{\mathbf{x}\in X^*} f(\mathbf{x}) - \min_{\mathbf{x}\in X^*} \check{f}_{X^*}(\mathbf{x}) = Kw(X^*)^\beta.$$

Therefore, the algorithm terminates only if $Kw(X^*)^\beta \leq \varepsilon$. □

The following arguments adopt the convention that a node $\tilde{X}$ is fathomed by value dominance only when $\min_{\mathbf{x}\in\tilde{X}} \check{f}_{\tilde{X}}(\mathbf{x}) > UBD_k$. In this situation, the stack is interpreted as representing the subset of C that can possibly contain global minima. This convention does not change the number of nodes processed by the branch-and-bound algorithm, it will only affect the number of nodes remaining on the stack at termination.

Lemma 2.2. *Define $\delta = \left(\frac{\varepsilon}{K}\right)^{\frac{1}{\beta}}$ and consider any node $\tilde{X} \in \mathbb{IC}$ with $w(\tilde{X}) \leq \delta$. Introduce the following partition of C:*

$$A = \{\mathbf{x} \in C : f(\mathbf{x}) - f(\mathbf{x}^*) > \varepsilon\},$$
$$B = \{\mathbf{x} \in C : f(\mathbf{x}) - f(\mathbf{x}^*) \leq \varepsilon\}.$$

Then, any node $\tilde{X} \subset A$ will be fathomed by value dominance.

Proof. Suppose that $\tilde{X} \subset A$ so that $\min_{\mathbf{x}\in\tilde{X}} f(\mathbf{x}) - UBD_k = \min_{\mathbf{x}\in\tilde{X}} f(\mathbf{x}) - f(\mathbf{x}^*) > \varepsilon$. By construction of $\tilde{X}$, even in the worst case, Eq. (2.1) implies that

$$\min_{\mathbf{x}\in\tilde{X}} f(\mathbf{x}) - \min_{\mathbf{x}\in\tilde{X}} \check{f}_{\tilde{X}}(\mathbf{x}) \leq K\delta^\beta = \varepsilon. \tag{2.3}$$

Since

$$\min_{\mathbf{x}\in\tilde{X}} \check{f}_{\tilde{X}}(\mathbf{x}) \geq \min_{\mathbf{x}\in\tilde{X}} f(\mathbf{x}) - \varepsilon > UBD_k$$

it follows that $\tilde{X}$ will be fathomed by value dominance. □

Note that this result indicates that any node $\tilde{X} \subset A$ will be fathomed when or before $w(\tilde{X}) = \delta$. On the other hand, consider a node $\tilde{X}$ such that $\tilde{X} \cap B \neq \emptyset$ and $w(\tilde{X}) = \delta$ with δ as defined in Lemma 2.2. From $\tilde{X} \cap B \neq \emptyset$,

$$\min_{\mathbf{x}\in\tilde{X}} f(\mathbf{x}) - UBD_k = \min_{\mathbf{x}\in\tilde{X}} f(\mathbf{x}) - f(\mathbf{x}^*) \leq \varepsilon$$

so that in the worst case (2.3)

$$\min_{\mathbf{x}\in\tilde{X}} \underset{\smile}{f}_{\tilde{X}}(\mathbf{x}) - UBD_k \leq 0.$$

In the worst case, such nodes will not be fathomed by value dominance.

Any node $\tilde{X}$ containing $\mathbf{x}^*$ will have $\tilde{X} \cap B \neq \emptyset$. In Lemma 2.1 it was argued that, when the convergence order bound is sharp, the node containing $\mathbf{x}^*$ must have width less than or equal to δ to guarantee termination. That is, in the worst case, B must be covered by nodes with $w(\tilde{X}) = \delta$ and none of them will be fathomed by value dominance.

2.1.1 Refinement of Neumaier's argument for a bound on the number of boxes necessary to cover B

Next, the number of boxes of width δ required to cover B is estimated. This argument will follow the idea presented by Neumaier [129]. Since f is twice differentiable at $\mathbf{x}^*$ and $\mathbf{x}^* \in \operatorname{int} C$, it follows that

$$f(\mathbf{x}) - f(\mathbf{x}^*) = \frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^{\mathrm{T}}\nabla^2 f(\mathbf{x}^*)(\mathbf{x}-\mathbf{x}^*) + r(\mathbf{x}-\mathbf{x}^*),$$

so that B is given by

$$B = \left\{ \mathbf{x} \in C : \frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^{\mathrm{T}}\nabla^2 f(\mathbf{x}^*)(\mathbf{x}-\mathbf{x}^*) + r(\mathbf{x}-\mathbf{x}^*) \leq \varepsilon \right\}. \tag{2.4}$$

Eq. (2.4) describes a nearly hyperellipsoidal region when $|r(\mathbf{x}-\mathbf{x}^*)| \ll \varepsilon$. This approximation becomes increasingly better for smaller ε because $|r(\mathbf{x}-\mathbf{x}^*)| \to 0$ as $\varepsilon \to 0$. Neumaier [129] compares the volume inside the hyperellipsoid, V, with the volume of a box to bound the number of boxes N that cover the interior of the hyperellipsoid from below. Denote $\Delta \equiv \det(\nabla^2 f(\mathbf{x}^*))$. Since

$$V(\varepsilon, n, \Delta) = \gamma_n \sqrt{\det\left[\left(\frac{\nabla^2 f(\mathbf{x}^*)}{2\varepsilon}\right)^{-1}\right]} = \gamma_n \sqrt{\frac{(2\varepsilon)^n}{\Delta}},$$

where $\gamma_n = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}$ [80], it follows that

$$N \approx \frac{V(\varepsilon, n, \Delta)}{\delta^n} = \frac{\gamma_n \sqrt{\frac{(2\varepsilon)^n}{\Delta}}}{\left(\frac{\varepsilon}{K}\right)^{\frac{n}{\beta}}} = \gamma_n K^{\frac{n}{\beta}} \sqrt{\frac{2^n}{\Delta}} \varepsilon^{n\left(\frac{1}{2}-\frac{1}{\beta}\right)}. \tag{2.5}$$

This argument is valid only when boxes are able to approximate the volume inside the hyperellipsoid well. Moreover, as $n \to \infty$, $\gamma_n \to 0$ [79, 80]. For constant Δ and ε, it follows that $V \to 0$ as $n \to \infty$. Thus, this argument suggests that the cluster problem should disappear for sufficiently large n for any fixed K, β, and Δ.

Consider the volume inside a slightly smaller hyperellipsoid by replacing ε in (2.4) with $\varepsilon - \xi$ where $0 < \xi \ll \varepsilon$. It is easy to show that

$$\frac{V(\varepsilon, n, \Delta) - V(\varepsilon - \xi, n, \Delta)}{V(\varepsilon, n, \Delta)} = 1 - \sqrt{\left(1 - \frac{\xi}{\varepsilon}\right)^n} \to 1 \quad \text{as} \quad n \to \infty.$$

For higher dimensions, nearly all of the volume inside the hyperellipsoid is close to its surface and thus it is also distributed in space (i.e., *not* concentrated at the center). The estimate for N, which was obtained by comparing volumes as suggested by (2.5), may not lead to accurate results. A different analysis is necessary.

2.1.2 A new analysis of the cluster problem

A new argument for the number of boxes of width δ required to cover the volume inside the hyperellipsoid B will be given. In particular, two cases will be considered here. First, the simpler case of a hypersphere (i.e., $\nabla^2 f(\mathbf{x}^*) = \mathbf{I}$) will be treated. Second, the results are then generalized to the case of a hyperellipsoid.

Assumption 2.3. Assume that there exists only one box $\tilde{X}$ visited by the branch-and-bound algorithm such that $w(\tilde{X}) = \delta$ and $\mathbf{x}^* \in \tilde{X}$. Furthermore, assume that $\mathbf{x}^*$ is in the center of $\tilde{X}$.

Note that if $\mathbf{x}^*$ is in the interior of the box, but not in the center, then it will become necessary to use an apparent box width $\delta' \equiv 2 \min_{i=1,\ldots,n} \{\tilde{x}_i^U - x_i^*, x_i^* - \tilde{x}_i^L\} \leq \delta$ in the following analysis instead.

For easier notation, a translated coordinate system $\mathbf{y} = \mathbf{x} - \mathbf{x}^*$ will be used hereafter, in which the considered approximation of B as the volume inside an hyperellipsoid is given by

$$\tilde{B} = \left\{ \mathbf{y} : \frac{1}{2\varepsilon} \mathbf{y}^{\mathrm{T}} \nabla^2 f(\mathbf{0}) \mathbf{y} \leq 1 \right\}.$$

Denote a *box* centered at $\mathbf{y}_0$ with width ω by $\square_\omega(\mathbf{y}_0) \equiv \{\mathbf{y} : \|\mathbf{y} - \mathbf{y}_0\|_\infty \leq \frac{\omega}{2}\}$.

Case 1: Hypersphere

Lemma 2.3. *Suppose that* $\nabla^2 f(\mathbf{x}^*) = \mathbf{I}$ *and let* $r = \sqrt{2\varepsilon}$.

(a) If $\delta \geq 2r$, *then let* $N = 1$.

(b) If $\frac{2r}{\sqrt{m-1}} > \delta \geq \frac{2r}{\sqrt{m}}$ *where* $m \in \mathbb{N}$, $m \leq n$, $2 \leq m \leq 18$, *then let*

$$N = \sum_{i=0}^{m-1} 2^i \binom{n}{i} + 2n \left\lceil \frac{m-9}{9} \right\rceil.$$

(c) *Otherwise let*

$$N = \left\lceil \frac{2r}{\delta\sqrt{2}} \right\rceil^{n-1} \left(\left\lceil \frac{2r}{\delta\sqrt{2}} \right\rceil + 2n \left\lceil \frac{r - \frac{r}{\sqrt{2}}}{\delta} \right\rceil \right).$$

Then, N is an upper bound on the number of boxes with width δ required to cover $\tilde{B}$.

Proof. By definition, $\tilde{B} = \left\{ \mathbf{y} : \frac{1}{2\varepsilon}\mathbf{y}^{\mathrm{T}}\mathbf{I}\mathbf{y} = \frac{1}{2\varepsilon}\mathbf{y}^{\mathrm{T}}\mathbf{y} \leq 1 \right\}$ describes the region inside a hypersphere about the origin with radius $r = \sqrt{2\varepsilon}$.

(a) Suppose that $\delta \geq 2r$. One finds immediately that $N = 1$ since $\mathbf{y} \in \tilde{B}$ implies $\mathbf{y} \in \square_\delta(\mathbf{0})$ as $\|\mathbf{y}\|_\infty \leq \|\mathbf{y}\|_2$ and $\delta \geq 2r$.

(b) Suppose that $1 < m \leq 18$ and $\delta \geq \frac{2r}{\sqrt{m}}$. Place a box with width δ at the center of the hypersphere. Let $\mathbf{e}_i$ be any n-vector whose components are 0 except i of the entries which are $\pm\frac{\delta}{2}$. Such an $\mathbf{e}_i$ represents the $(n-i)$-faces of the hypercube. In particular, each $\mathbf{e}_i$ is the midpoint of such a face and it is well known that an n-dimensional hypercube has $F(n,i) \equiv 2^i\binom{n}{i}$ of these. Hence, $\mathbf{e}_i$ is representative of $F(n,i)$ directions.

It will be argued that, in addition to the central box, placing a single box along each of the $\mathbf{e}_1, \ldots, \mathbf{e}_m$ directions is sufficient to cover $\tilde{B}$.

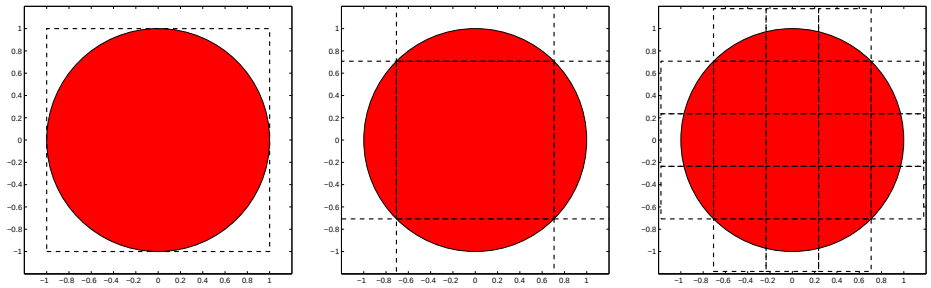
If $\delta > \frac{2r}{\sqrt{m}}$ then $\frac{2r}{\delta\sqrt{m}}\mathbf{e}_m \notin \tilde{B}$. If $\delta = \frac{2r}{\sqrt{m}}$, then $\frac{2r}{\delta\sqrt{m}}\mathbf{e}_m \in \square_\delta(\mathbf{0})$ and also $\frac{2r}{\delta\sqrt{m}}\mathbf{e}_m \in \partial\tilde{B}$. As a consequence, faces lower than the $(n-m)$-face need not be considered as they do not intersect the hypersphere whereas additional boxes must be placed in the direction of all faces from the $(n-m+1)$-face up to the $(n-1)$-face to cover $\tilde{B}$.

Set $\delta = \frac{2r}{\sqrt{m}}$, the width of the smallest permissible box. Next, consider the shortest distance from $\frac{2r}{\delta\sqrt{i}}\mathbf{e}_i$, which is a point on the surface of the hypersphere, to the surface of the central box in the ∞-norm: $\frac{r}{\sqrt{i}} - \frac{\delta}{2} = \frac{r}{\sqrt{i}} - \frac{r}{\sqrt{m}}$. When this distance is smaller than δ, then one box suffices to cover the remaining parts of the hypersphere in the $\mathbf{e}_i$ direction. This holds true for any $i = 2, \ldots, m$ and $m \leq 18$. When $m > 9$, then two boxes must be placed in each of the $\mathbf{e}_1$ directions, however.

(c) Otherwise, the central region inside the hypersphere cannot be covered by a single box. Instead, a number of boxes that grows exponentially with n is necessary to fill the central region. Additional boxes must be placed along the coordinate axes so that $N = m^n + 2nm^{n-1}\left\lceil \frac{1}{\delta}\left(r - \frac{r}{\sqrt{2}}\right)\right\rceil$ where m is the smallest integer so that $m\delta \geq \frac{2r}{\sqrt{2}}$. Thus, the result follows.

□

Note that the number of boxes presented for the second case in Lemma 2.3 is $O(n^{m-1})$. Also, while it is possible to construct tighter bounds on N for the case $m > 18$, it becomes much more involved. In particular, it becomes necessary to place more than one box in the



(a) (b) (c)

Figure 2.2: Illustration of different cases for a circle where dashed regions show boxes required to cover $\tilde{B}$

$\mathbf{e}_i$, $i > 1$ direction, which complicates the geometry. Roughly speaking, these boxes will only touch each other in a lower dimensional face leaving parts of $\tilde{B}$ uncovered, hence, requiring additional boxes.

The different cases are illustrated for $n = 2$ and $m = 2$ in Figure 2.2.

Case 2: Hyperellipsoid

The results in Lemma 2.3 will now be generalized to a hyperellipsoid by dropping the assumption that $\nabla^2 f(\mathbf{x}^*) = \mathbf{I}$.

Theorem 2.1. *Let $\lambda_1 > 0$ be the smallest eigenvalue of $\nabla^2 f(\mathbf{x}^*)$ and $r = \sqrt{\frac{2\varepsilon}{\lambda_1}}$.*

(a) If $\delta \geq 2r$ or, equivalently, if

$$\left(\frac{\varepsilon}{K}\right)^{\frac{1}{\beta}} \geq 2\sqrt{\frac{2\varepsilon}{\lambda_1}},$$

then let $N = 1$.

(b) Suppose that $\frac{2r}{\sqrt{m-1}} > \delta \geq \frac{2r}{\sqrt{m}}$ where $m \in \mathbb{N}$, $m \leq n$, $2 \leq m \leq 18$ or, equivalently,

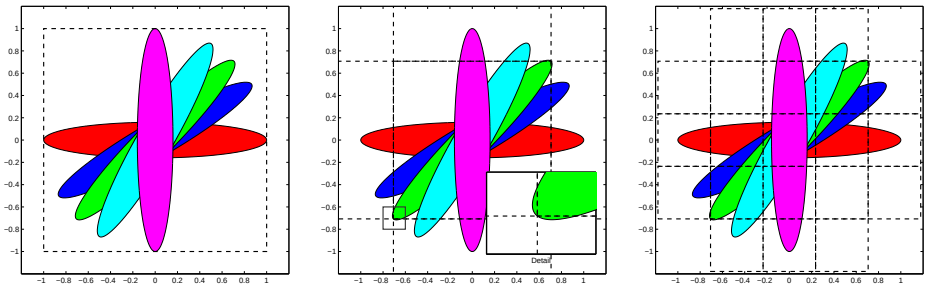
$$\frac{2\sqrt{2\varepsilon}}{\sqrt{(m-1)\lambda_1}} > \left(\frac{\varepsilon}{K}\right)^{\frac{1}{\beta}} \geq \frac{2\sqrt{2\varepsilon}}{\sqrt{m\lambda_1}}.$$

Then let

$$N = \sum_{i=0}^{m-1} 2^i \binom{n}{i} + 2n \left\lceil \frac{m-9}{9} \right\rceil.$$

(c) Otherwise, let

$$N = \left\lceil 2K^{\frac{1}{\beta}} \varepsilon^{(\frac{1}{2}-\frac{1}{\beta})} \lambda_1^{-\frac{1}{2}} \right\rceil^{n-1} \left(\left\lceil 2K^{\frac{1}{\beta}} \varepsilon^{(\frac{1}{2}-\frac{1}{\beta})} \lambda_1^{-\frac{1}{2}} \right\rceil + 2n \left\lceil (\sqrt{2}-1) K^{\frac{1}{\beta}} \varepsilon^{(\frac{1}{2}-\frac{1}{\beta})} \lambda_1^{-\frac{1}{2}} \right\rceil \right).$$



(a) (b) (c)

Figure 2.3: Illustration of different cases for an ellipse where dashed regions show boxes required to cover $\tilde{B}$

Then, N is an upper bound on the number of boxes with width δ required to cover $\tilde{B}$.

Proof. Suppose $\mathbf{y} \in \tilde{B}$ so that $\mathbf{y}^{\mathrm{T}} \nabla^2 f(\mathbf{0}) \mathbf{y} \leq 2\varepsilon$. By Rayleigh's principle,

$$0 \leq \lambda_1 \mathbf{y}^{\mathrm{T}} \mathbf{y} \leq \mathbf{y}^{\mathrm{T}} \nabla^2 f(\mathbf{0}) \mathbf{y} \leq 2\varepsilon$$

so that $\lambda_1 \mathbf{y}^{\mathrm{T}} \mathbf{y} \leq 2\varepsilon$. Thus, $\tilde{B} \subset \left\{ \mathbf{z} : \frac{\lambda_1}{2\varepsilon} \mathbf{z}^T \mathbf{z} \leq 1 \right\}$, the volume inside a hypersphere with radius $r = \sqrt{2\varepsilon \lambda_1^{-1}}$. Hence, Lemma 2.3 with $r = \sqrt{2\varepsilon \lambda_1^{-1}}$ can be applied and it provides an upper bound for N. □

The different cases are illustrated for $n = 2$ and $m = 2$ in Figure 2.3.

2.2 Discussion of Theorem 2.1

Studying the expressions derived above for N, two characteristics are noteworthy:

- the variation of N with ε depends on the value of β and
- the influence of K on the behavior of N.

Both will be discussed in more detail. First, consider the functional dependence of N on ε for different β.

1. When $\beta = 1$ and K not necessarily small, then $N \propto \left(\frac{1}{\varepsilon}\right)^{\frac{n}{2}}$. The number of boxes required to cover the hyperellipsoid will grow rapidly as the convergence tolerance ε is decreased—the well-known cluster problem.
2. When $\beta = 2$, then N is independent of ε for any value of K, i.e., the number of boxes required to cover the hyperellipsoid is insensitive to ε.

Case	N
$K \leq \frac{\lambda_1}{8}$	1
$\frac{\lambda_1}{8} < K \leq \frac{\lambda_1}{4}$	$1 + 2n$
$\frac{\lambda_1}{4} < K \leq \frac{3\lambda_1}{8}$	$1 + 2n^2$
$\frac{3\lambda_1}{8} < K \leq \frac{\lambda_1}{2}$	$1 + \frac{8}{3}n - 3n^2 + \frac{4}{3}n^3$
$\vdots$	$\vdots$
$K > \frac{9\lambda_1}{4}$	$\left\lceil 2\sqrt{K\lambda_1^{-1}} \right\rceil^{n-1} \left(\left\lceil 2\sqrt{K\lambda_1^{-1}} \right\rceil + 2n \left\lceil (\sqrt{2}-1)\sqrt{K\lambda_1^{-1}} \right\rceil \right)$

Table 2.2: Summary of results for number of boxes required to cover $\tilde{B}$ when $\beta = 2$

3. When $\beta = 3$ and K not necessarily small, then $N \propto \varepsilon^{\frac{n}{6}}$, and the number of boxes required to cover the hyperellipsoid will decrease with decreasing ε. Note that this does not necessarily mean that the total number of nodes required for termination decreases with the tolerance, because this analysis only estimates the number of nodes to cover B, which itself decreases in size as ε is decreased.

These observations agree with the results found in the literature [54, 129].

Second, assume that $\beta = 2$ and focus on how K parametrizes the behavior of N. Table 2.2 summarizes these results for the case $\beta = 2$. When K is sufficiently small, i.e., $K \leq \frac{\lambda_1}{8}$, the cluster problem is completely absent ($N = 1$). Recall that λ_1 denotes the smallest eigenvalue of $\nabla^2 f(\mathbf{x}^*)$. When λ_1 is small, this bound may only hold for $K \ll 1$. Depending on the magnitude of K, N is polynomial (of varying degree) in n. For example, when $K \leq \frac{\lambda_1}{4}$, then N grows linearly with problem size and when $K \leq \frac{3\lambda_1}{8}$ the number of boxes grows quadratically with n. Both cases are remarkable as they suggest a fundamentally different behavior of different relaxations with second-order convergence each depending on these thresholds for K. Prior analyses of the cluster problem [54, 129] stop short of explicitly drawing this conclusion.

2.3 Cluster problem for problems with non-differentiable functions

Whereas the previous analysis and discussion focused on clustering occurring in the vicinity of an unique unconstrained minimum of a twice differentiable function, here, the differentiability assumption will be removed. It will be studied if the cluster problem occurs in this setting, too. In particular, points of non-differentiability will be analyzed next as otherwise a neighborhood of the minimum exists in which the function is differentiable.

Assumption 2.4. Suppose $D \subset \mathbb{R}^n$ and let $f : D \to \mathbb{R}$. Suppose $C \subset D$ is convex and $\mathbf{x}^*$ is a global minimizer on C.

Note that no assumption regarding the regularity of f is made, in contrast to Assumption 2.1, which is not assumed to hold in this Section.

It is easy to see that Lemma 2.1 still holds and, again, it suffices to look at

$$B = \{\mathbf{x} \in C : f(\mathbf{x}) - f(\mathbf{x}^*) \leq \varepsilon\},$$

the set of points in C for which fathoming by value dominance cannot be guaranteed, cf. Lemma 2.2.

As before, we want to characterize B conservatively.

Lemma 2.4. *Let* conv B *denote the convex hull of* B *and let* $\check{f}$ *denote the convex envelope of* f *on* conv B. *Set* $L = \min_{\|\mathbf{d}\|_1=1} \max_{\sigma \in \partial_B \check{f}(x^*)} \sigma^{\mathrm{T}}\mathbf{d}$ *where* $\partial_B \check{f}(\mathbf{x}^*)$ *denotes the Bouligand differential of* $\check{f}$ *at* $\mathbf{x}^*$. *Then, we can approximate* B *conservatively as*

$$\check{B} \equiv \{\mathbf{x} \in C : L\|\mathbf{x} - \mathbf{x}^*\|_1 \leq \varepsilon\} \supset B. \tag{2.6}$$

Proof. Convexity of $\check{f}$ and the definition of $\partial_B \check{f}$ imply that for all $\mathbf{x} \in$ conv B and for any $\sigma \in \partial_B \check{f}(\mathbf{x}^*)$

$$\sigma^{\mathrm{T}}(\mathbf{x} - \mathbf{x}^*) \leq \check{f}(\mathbf{x}) - \check{f}(\mathbf{x}^*). \tag{2.7}$$

In the following, let $\mathbf{d} \in \mathbb{R}^n$ so that $\|\mathbf{d}\|_1 = 1$. We want to identify the direction $\mathbf{d}$ in which f grows the slowest, i.e., a $\mathbf{d}$ that minimizes $\check{f}'(\mathbf{x}^*, \mathbf{d})$, the directional derivative of $\check{f}$ at $\mathbf{x}^*$. An equivalent characterization, cf. [82, p. 345], is

$$L = \min_{\|\mathbf{d}\|_1=1} \max_{\sigma \in \partial_B \check{f}(x^*)} \sigma^{\mathrm{T}}\mathbf{d}.$$

Since $\mathbf{x}^*$ is a minimizer, $\check{f}'(\mathbf{x}^*, \mathbf{d}) \geq 0$ for all $\mathbf{d} \in \mathbb{R}^n$ [82, Theorem VI-2.2.1] so that $L \geq 0$.

Suppose that $\mathbf{x} \in B$. Thus, there exists a $t \geq 0$ and a $\mathbf{d} \in \mathbb{R}^n$, $\|\mathbf{d}\|_1 = 1$ so that $t\mathbf{d} = \mathbf{x} - \mathbf{x}^*$. Also,

$$\check{f}(\mathbf{x}^* + t\mathbf{d}) \leq f(\mathbf{x}^* + t\mathbf{d}) \leq f(\mathbf{x}^*) + \varepsilon. \tag{2.8}$$

From the definition of L and (2.7), we have

$$Lt \leq \check{f}(\mathbf{x}^* + t\mathbf{d}) - \check{f}(\mathbf{x}^*). \tag{2.9}$$

Since C is convex and $B \subset C$, it follows that conv $B \subset C$. Since $\mathbf{x}^* \in B$ minimizes $\mathbf{f}$ on C, and thus also on conv B, and since $\check{f}$ is the convex envelope of f, $\check{f}(\mathbf{x}^*) = \mathbf{f}(\mathbf{x}^*)$. Combine (2.8) and (2.9) and note that $\|\mathbf{x} - \mathbf{x}^*\|_1 = t$ to see that $\mathbf{x} \in \check{B}$. □

Whereas Lemma 2.3 was applied to $\tilde{B}$ characterized by the 2-norm, $\check{B}$ is described by the 1-norm. However, the analysis carries over after accounting for this change with $L \neq 0$ and we used $r = \frac{\varepsilon}{L}$ instead. In Case (b), we simply need to change the condition from $\delta \geq \frac{2r}{\sqrt{m}}$ to $\delta \geq \frac{2r}{m}$ and we need to require $1 < m \leq 6$ instead of $1 < m \leq 18$. In Case (c), we can simply replace $\sqrt{2}$ by 2.

Corollary 2.1. Suppose that $L \neq 0$. Let $r = \frac{\varepsilon}{L}$.

(a) If $\delta \geq 2r$ or, equivalently, if

$$\left(\frac{\varepsilon}{K}\right)^{\frac{1}{\beta}} \geq 2\frac{\varepsilon}{L},$$

then let $N = 1$.

(b) Suppose that $\frac{2r}{m-1} > \delta \geq \frac{2r}{m}$ where $m \in \mathbb{N}$, $m \leq n$, $2 \leq m \leq 6$ or, equivalently,

$$\frac{2\varepsilon}{(m-1)L} > \left(\frac{\varepsilon}{K}\right)^{\frac{1}{\beta}} \geq \frac{2\varepsilon}{mL}.$$

Then let

$$N = \sum_{i=0}^{m-1} 2^i \binom{n}{i} + 2n \left\lceil \frac{m-3}{3} \right\rceil.$$

(c) Otherwise, let

$$N = \left\lceil K^{\frac{1}{\beta}} \varepsilon^{(1-\frac{1}{\beta})} L^{-1} \right\rceil^{n-1} \left(\left\lceil K^{\frac{1}{\beta}} \varepsilon^{(1-\frac{1}{\beta})} L^{-1} \right\rceil + 2n \left\lceil \frac{1}{2} K^{\frac{1}{\beta}} \varepsilon^{(1-\frac{1}{\beta})} L^{-1} \right\rceil \right).$$

Then, N is an upper bound on the number of boxes with width δ required to cover $\tilde{B}$.

Remark 2.1. Note that, in contrast to the results in Section 2.2, the explicit dependence of N on ε disappears already for $\beta = 1$. Therefore, even for *relaxations with linear convergence order only*, N will not increase exponentially for decreasing termination tolerance. The discussion of the dependence of N on K that originally considered $\beta = 2$ is valid for $\beta = 1$ now, cf. Table 2.2. Therefore, fundamentally different behavior of different relaxations with first-order convergence depending on these thresholds for K is expected.

Remark 2.2. Consider the following cases where $L = 0$ so that Corollary 2.1 is not applicable.

- If $\mathbf{f}$ is differentiable on D and $\mathbf{x}^*$ is the unique minimizer on C so that $\nabla f(\mathbf{x}^*) = 0$ then $L = 0$. This case, however, is covered by the analysis in Section 2.1.2.
- If there are multiple minimizers of $\mathbf{f}$ on C, regardless of the regularity of $\mathbf{f}$, then $L = 0$. While for functions with a finite number of minimizers, the analysis can be applied to sufficiently small neighborhoods of each minimizer on which $L \neq 0$, this is not possible if the set of minimizers is connected. Examples of this case include parameter estimation problems where parameters are not identifiable and an infinite number of optimal solutions exists.
- If $\mathbf{f}$ is convex in a neighborhood of $\mathbf{x}^*$ and there exists a direction, in which the directional derivative of $\mathbf{f}$ at $\mathbf{x}^*$ is zero, then $L = 0$. $f(x) = \max\{x^2, x\}$ with $x^* = 0$ is an example of this case.

Examples of functions, for which Corollary 2.1 is applicable, include cases such as $f(x) = |x|$ or $f(\mathbf{x}) = \max\{f^1(\mathbf{x}), f^2(\mathbf{x})\}$ where f^1, f^2 are convex and the directional derivative of f^1, f^2 at $\mathbf{x}^*$ is nonzero for any direction.

2.4 Conclusion

The analysis of the cluster problem has been revisited in this chapter. Prior results that reveal the dependence of the cluster problem on the convergence order β and the termination tolerance ε have been verified. Furthermore, even for relaxations with $\beta = 2$, the new analysis indicates fundamentally different scaling behavior depending on the value of K, the pre-factor in the convergence order. Thus, tighter relaxations can lead to dramatic improvements in mitigating the cluster problem.

When the objective function is not differentiable at the minimum, then linearly convergent relaxations are sufficient to avoid the cluster problem. The new analysis shows also in this case that different regimes exists depending on K.

Chapter 3

Factorable functions and methods to bound their range

The notion of a factorable function is central for the remainder of this thesis. Early references to the concept of the factorable function include [118, 136]. Conceptually, a factorable function is any function that can be represented finitely on a computer without resorting to `IF` or `WHILE` statements. For this particular class of functions, several bounding methods have been developed previously. In this chapter, the focus lies on methods based on interval analysis [122, 127] and McCormick's composition result [118, 156], αBB relaxations [1, 5] are also briefly mentioned. In particular, we will also report results on the convergence order of the methods. In this chapter, we will mostly follow the notation developed in [155].

3.1 Concept of factorable functions

In this section, we will formalize the definition of a factorable function and also show its representation with a directed acyclic graph. Loosely speaking, a function is factorable if it can be represented as a finite sequence of simple binary operations and univariate functions.

3.1.1 Basic definition

Hereafter, a function will be denoted as a triple (o, B, R) where B is the domain, R is the range, and o is a mapping from B into R, $o : B \to R$. Permissible functions shall include binary addition $(+, \mathbb{R}^2, \mathbb{R})$ and multiplication $(\times, \mathbb{R}^2, \mathbb{R})$ as well as a collection of univariate functions, cf. Definition 3.1.

Definition 3.1. Let $\mathcal{L}$ denote a set of functions $(u, B, \mathbb{R})$ where $B \subset \mathbb{R}$. $\mathcal{L}$ will be referred to as a *library of univariate functions*.

It will be required that, for each $(u, B, \mathbb{R}) \in \mathcal{L}$, $u(x)$ can be evaluated on a computer for any $x \in B$. Additional assumptions will be introduced when necessary.

Without loss of generality, we can also consider the binary operations $(-, \mathbb{R}^2, \mathbb{R})$ and $(/, \mathbb{R} \times \mathbb{R}\backslash\{0\}, \mathbb{R})$, which are contained in the framework discussed above by combining the univariate functions $u(x) = -x$ or $u(x) = \frac{1}{x}$ with the binary addition or multiplication,

respectively. Sometimes more efficient calculations are possible though when subtraction and division are considered directly.

Factorable functions will be defined in terms of *computational sequences*, which are ordered sequences of the permissible basic operations defined above. Every such sequence of computations defines a sequence of intermediate quantities called *factors*. In the following definition, the factors are denoted by v_k, and the functions π_k are used to select one or two previous factors to be the operand(s) of the next operation. Note that a computational sequence is a specialization of a DAG because it allows binary and unary operations only.

Definition 3.2. Let $n_i, n_o \geq 1$. A *$\mathcal{L}$-computational sequence* with n_i inputs and n_o outputs is a pair $(\mathcal{S}, \pi_o)$, where:

1. $\mathcal{S}$ is a finite sequence of pairs $\{((o_k, B_k, \mathbb{R}), (\pi_k, \mathbb{R}^{k-1}, \mathbb{R}^{d_k}))\}_{k=n_i+1}^{n_f}$ with every element defined by one of the following options:
   a) $(o_k, B_k, \mathbb{R})$ is either $(+, \mathbb{R}^2, \mathbb{R})$ or $(\times, \mathbb{R}^2, \mathbb{R})$ and $\pi_k : \mathbb{R}^{k-1} \to \mathbb{R}^2$ is defined by $\pi_k(\mathbf{v}) = (v_i, v_j)$ for some integers $i, j \in \{1, \ldots, k-1\}$.
   b) $(o_k, B_k, \mathbb{R}) \in \mathcal{L}$ and $\pi_k : \mathbb{R}^{k-1} \to \mathbb{R}$ is defined by $\pi_k(\mathbf{v}) = v_i$ for some integer $i \in \{1, \ldots, k-1\}$.
2. $\pi_o : \mathbb{R}^{n_f} \to \mathbb{R}^{n_o}$ is defined by $\pi_o(\mathbf{v}) = (v_{i(1)}, \ldots, v_{i(n_o)})$ for some integers $i(1), \ldots, i(n_o) \in \{1, \ldots, n_f\}$.

A computational sequence defines a function $\mathbf{f}_\mathcal{S} : D_\mathcal{S} \subset \mathbb{R}^{n_i} \to \mathbb{R}^{n_o}$ by the following construction.

Definition 3.3. Let $(\mathcal{S}, \pi_o)$ be a $\mathcal{L}$-computational sequence with n_i inputs and n_o outputs. Define the *sequence of factors* $\{(v_k, D_k, \mathbb{R})\}_{k=1}^{n_f}$ with $D_k \subset \mathbb{R}^{n_i}$, where

1. for $k = 1, \ldots, n_i$, $D_k = \mathbb{R}^{n_i}$ and $v_k(\mathbf{x}) = x_k$, $\forall \mathbf{x} \in D_k$,
2. for $k = n_i + 1, \ldots, n_f$, $D_k = \{\mathbf{x} \in D_{k-1} : \pi_k(v_1(\mathbf{x}), \ldots, v_{k-1}(\mathbf{x})) \in B_k\}$ and $v_k(\mathbf{x}) = o_k(\pi_k(v_1(\mathbf{x}), \ldots, v_{k-1}(\mathbf{x})))$, $\forall \mathbf{x} \in D_k$.

The set $D_\mathcal{S} \equiv D_{n_f}$ is the *natural domain* of $(\mathcal{S}, \pi_o)$, and the *natural function* $(\mathbf{f}_\mathcal{S}, D_\mathcal{S}, \mathbb{R}^{n_o})$ is defined by $\mathbf{f}_\mathcal{S}(\mathbf{x}) = \pi_o(v_1(\mathbf{x}), \ldots, v_{n_f}(\mathbf{x}))$, $\forall \mathbf{x} \in D_\mathcal{S}$.

Definition 3.4. A function $\mathbf{f} : D \subset \mathbb{R}^{n_i} \to \mathbb{R}^{n_o}$ will be called *$\mathcal{L}$-factorable* if there exists a $\mathcal{L}$-computational sequence $(\mathcal{S}, \pi_o)$ with n_i inputs and n_o outputs such that the natural function $(\mathbf{f}_\mathcal{S}, D_\mathcal{S}, \mathbb{R}^{n_o})$ satisfies $D \subset D_\mathcal{S}$ and $\mathbf{f} = \mathbf{f}_\mathcal{S}|_D$.

When we refer to as a function as $\mathcal{L}$-factorable, we implicitly assume that Assumptions 3.1 and 3.3 hold. In certain cases, we also need Assumptions 3.2 and 3.4 to hold. This will be noted where necessary.

k	o_k	π_k
1	x_1	
2	x_2	
3	x_3	
4	$(\cdot)^3$	$\pi(\mathbf{v}) = (v_2)$
5	$-$	$\pi(\mathbf{v}) = (v_1, v_2)$
6	$\log(\cdot)$	$\pi(\mathbf{v}) = (v_5)$
7	$\times$	$\pi(\mathbf{v}) = (v_4, v_6)$
8	$+$	$\pi(\mathbf{v}) = (v_1, v_7)$
9	$\times 0.25$	$\pi(\mathbf{v}) = (v_3)$
10	$\exp(\cdot)$	$\pi(\mathbf{v}) = (v_9)$
11	$+$	$\pi(\mathbf{v}) = (v_2, v_{10})$
12	$\times$	$\pi(\mathbf{v}) = (v_5, v_{11})$
13	$(\cdot)^2$	$\pi(\mathbf{v}) = (v_5)$
14	$-$	$\pi(\mathbf{v}) = (v_4, v_{13})$

Table 3.1: One possible representation of $\mathbf{f}$ in Example 3.1 as a $\mathcal{L}$-computational sequence.

3.1.2 Representation as directed acyclic graph

Factorable functions can be represented as a directed acyclic graph. As its name implies, a directed acyclic graph is a collection of vertices and directed edges. Vertices are connected by these edges in such a way so that it is impossible to start at any arbitrary edge, to travel from vertex to vertex along the direction of the edge and to return to the starting point [43].

Example 3.1. Let $D = \{\mathbf{x} \in \mathbb{R}^3 : x_1 > x_2\}$. Consider $\mathbf{f} : D \to \mathbb{R}^3$ given by

$$\begin{aligned} f_1(\mathbf{x}) &= x_1 + \left(x_2^3 \ln(x_1 - x_2)\right), \\ f_2(\mathbf{x}) &= x_2^3 - (x_1 - x_2)^2, \\ f_3(\mathbf{x}) &= (x_1 - x_2)\left(x_2 + \exp(0.25 x_3)\right). \end{aligned}$$

One possible representation of $\mathbf{f}$ as a $\mathcal{L}$-computational sequence is given in Table 3.1 with $\pi_o(\mathbf{v}) = (v_8, v_{12}, v_{14})$. The directed acyclic graph corresponding to the computational sequence is shown in Figure 3.1.

3.2 Interval analysis

Interval analysis was first introduced in the 1960s by Moore [122]. It has been used in a variety of applications, but the main theme is either verifying floating-point calculations performed with finite precision to account for possible round-off error or bounding the range of a function on a domain. The main advantage of interval-based calculations is

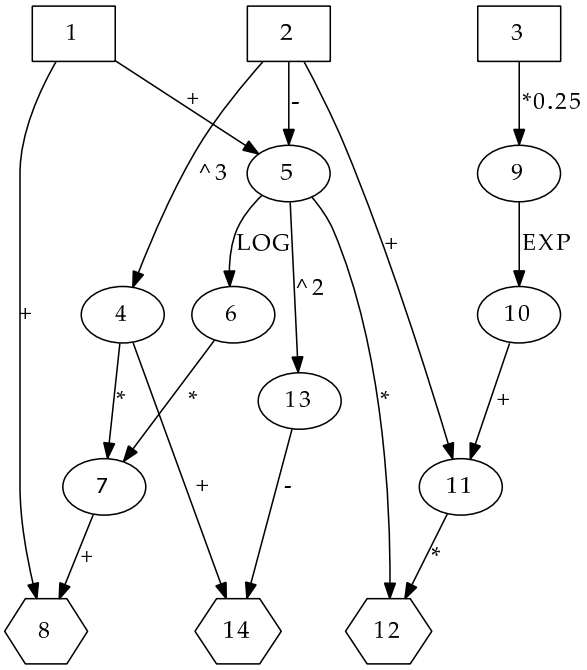


Figure 3.1: Directed acyclic graph of the $\mathcal{L}$-computational sequence given in Table 3.1 for the $\mathcal{L}$-factorable function $\mathbf{f}$ in Example 3.1

their small computational overhead, which is obtained at the expense of weak bounds when used with wide intervals. The reader is referred to [3, 122, 127] for reviews of the interval literature. Below, we will collect definitions and results necessary for this thesis.

Definition 3.5. For $a, b \in \mathbb{R}$, $a \leq b$ define the *interval* $[a, b]$ as the compact, connected set $\{x \in \mathbb{R} : a \leq x \leq b\}$. The set of all nonempty intervals is denoted as $\mathbb{IR}$, and intervals are denoted by capital letters, $X \in \mathbb{IR}$. The set of n-dimensional intervals (Cartesian products of n intervals) is denoted by $\mathbb{IR}^n$. Suppose $X \in \mathbb{IR}^n$. Then, the *lower* and *upper bounds* of X are denoted as $\underline{\mathbf{x}}$ and $\overline{\mathbf{x}}$, respectively. Suppose $Z \subset \mathbb{R}^n$. The set of all interval subsets of Z is denoted by $\mathbb{IZ} \subset \mathbb{IR}^n$. Lastly, if Z is nonempty and bounded, then $\text{hull}(Z)$ or $\square Z$ with $(\text{hull}\, Z)_i = \square Z_i = [\inf_{\mathbf{z} \in Z} z_i, \sup_{\mathbf{z} \in Z} z_i]$, $i = 1, \ldots, n$ denotes *interval hull* of Z, the tightest interval enclosing Z.

Note that $(\cdot)^L$ and $(\cdot)^U$ will be used in some instances for more complex expressions to denote the respective lower and upper bound of an interval.

Definition 3.6. Suppose $X, Y \in \mathbb{IR}^n$. The *midpoint* of X is denoted by $m(X) = \frac{1}{2}(\underline{\mathbf{x}} + \overline{\mathbf{x}})$. The *width* of X is denoted by $w(X) = \max_i\{\overline{x}_i - \underline{x}_i\}$. The *absolute value* of X is denoted by $|X| = (|X_1|, \ldots, |X_n|)$ where $|X_i| = \max\{|\underline{x}_i|, |\overline{x}_i|\}$. Denote the *Haussdorff metric* by $d_H(X, Y) = \max_i \max\{|\underline{x}_i - \underline{y}_i|, |\overline{x}_i - \overline{y}_i|\}$.

It is easy to show that d_H defines a metric on $\mathbb{IR}^n$ [e.g., 127, 1.7.2]. A sequence of intervals $\{X^k\}$ converges to X^* if $\lim_{k \to \infty} d_H(X^k, X^*) = 0$.

We will encounter functions that either return a vector of reals or the symbol NaN, or "not a number", which can be thought of as undefined or unspecified. It is convenient to define $\mathbb{R}_\emptyset = \mathbb{R} \cup \{\text{NaN}\}$. For the purposes of this thesis it is also necessary to extend the definition of an interval to include unbounded intervals and empty intervals, which are commonly excluded in the definition of $\mathbb{IR}$ [e.g, 95]. Here, $\emptyset$ is used to denote the empty interval.

Definition 3.7. Let $\mathbb{IR}_\emptyset \equiv \mathbb{IR} \cup \{\emptyset\}$ and let the set of all interval subsets of $Z \subset \mathbb{R}^n$ including the empty interval be denoted by $\mathbb{I}_\emptyset Z \subset \mathbb{IR}_\emptyset^n$. Similarly to Definition 3.5, define the *set of all extended intervals* as $\bar{\mathbb{IR}} = \{[a,b] : a,b \in \mathbb{R} \cup \{+\infty, -\infty\}, a \leq b\} \cup \{\emptyset\}$, which includes all unbounded intervals and also the empty interval. Lastly, the set of all extended interval subsets of $Z \subset \mathbb{R}^n$ is denoted by $\bar{\mathbb{I}}Z \subset \bar{\mathbb{IR}}^n$.

We will follow the conventions that real-valued operations involving NaN result in NaN, that $[\text{NaN}, \text{NaN}] = \emptyset$, that NaN is an element of any interval, that every interval contains the empty interval and that any interval operation involving the empty interval will again result in the empty interval with the exception of the construction of the interval hull where $\text{hull}(\{X, \emptyset\}) = X$ for any $X \in \bar{\mathbb{IR}}^n$. Note that $X = \emptyset$ for $X \in \bar{\mathbb{IR}}^n$ if $X_i = \emptyset$ for some $i = 1, \ldots, n$. Otherwise, the operations of interval arithmetic extend naturally. For any $x \in \mathbb{R}$ and $\circ \in \{+, -, \cdot, /\}$, define $x \circ \pm\infty = \lim_{a \to \pm\infty} x \circ a$.

Definition 3.8. Let $\mathbf{f} : D \subset \mathbb{R}^n \to \mathbb{R}_\emptyset^m$, and for any $E \subset D$, let $\mathbf{f}(E)$ denote the image of E under $\mathbf{f}$. A mapping $F : \mathfrak{D} \subset \bar{\mathbb{I}}D \to \bar{\mathbb{IR}}^m$ is an *inclusion function* for $\mathbf{f}$ on $\mathfrak{D}$ if $\mathbf{f}(X) \subset F(X)$, $\forall X \in \mathfrak{D}$.

While the concept of an inclusion function is very relevant to global optimization, a simpler construction can yield this property as will be shown below.

Definition 3.9. Let $D \subset \mathbb{R}^n$. A set $\mathfrak{D} \subset \bar{\mathbb{IR}}^n$ is an *interval extension* of D if $\mathfrak{D} \subset \bar{\mathbb{I}}D$ and every $\mathbf{x} \in D$ satisfies $[\mathbf{x}, \mathbf{x}] \in \mathfrak{D}$. Let $\mathbf{f} : D \to \mathbb{R}_\emptyset^m$. A function $F : \mathfrak{D} \subset \bar{\mathbb{I}}D \to \bar{\mathbb{IR}}^m$ is an *interval extension* of $\mathbf{f}$ on D if $\mathfrak{D}$ is an interval extension of D and $F([\mathbf{x}, \mathbf{x}]) = [\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x})]$ for every $\mathbf{x} \in D$.

Definition 3.10. Let $F : \mathfrak{D} \subset \bar{\mathbb{IR}}^n \to \bar{\mathbb{IR}}^m$. F is *inclusion monotonic* on $\mathfrak{D}$ if

$$X_1 \subset X_2 \Rightarrow F(X_1) \subset F_2(X), \quad \forall X_1, X_2 \in \mathfrak{D}.$$

Theorem 3.1. *Let $\mathbf{f} : D \subset \mathbb{R}^n \to \mathbb{R}_\emptyset^m$ and let $F : \mathfrak{D} \to \bar{\mathbb{IR}}^m$ be an interval extension of $\mathbf{f}$. If F is inclusion monotonic on $\mathfrak{D}$, then F is an inclusion function of $\mathbf{f}$ on $\mathfrak{D}$.*

Proof. Choose any $X \in \mathfrak{D}$ and any $\mathbf{x} \in X$. Since $\mathbf{x} \in D$, it follows that $[\mathbf{x}, \mathbf{x}] \in \mathfrak{D}$. Since $\emptyset \in F(X)$ is always true, if $f(\mathbf{x}) = \emptyset$ then $f(\mathbf{x}) \in F(X)$. Otherwise, the result follows from [155, Theorem 2.3.4]. □

Definition 3.11. Let $F : \mathfrak{D} \subset \mathbb{IR}^n \to \mathbb{IR}^m$. F is *locally Lipschitz* on $\mathfrak{D}$ if for every $\tilde{X} \in \mathfrak{D}$ there exist $\delta, L > 0$ such that $d_H(F(X), F(Y)) \leq L d_H(X, Y)$, $\forall X, Y \in \{Z \in \mathfrak{D} : d_H(Z, \tilde{X}) < \delta\}$.

Theorem 3.2. *Suppose $F : \mathfrak{D} \subset \mathbb{IR}^n \to \mathbb{IR}^m$ is locally Lipschitz on $\mathfrak{D}$. Then, there exists $L > 0$ so that $w(F(X)) \leq Lw(X)$, $\forall X \in \mathfrak{D}$.*

Proof. Let λ_f be as defined in [127, 2.1.2]. Set $L = \max_i \lambda_{f,i}$ and the result follows from [127, 2.1.2]. □

Lemma 3.1. *Let $\mathbf{f} : D \subset \bar{\mathbb{R}}^n \to \mathbb{R}^m$ and let $F : \mathfrak{D} \to \mathbb{IR}_\emptyset^m$ be an inclusion function of $\mathbf{f}$. If F is locally Lipschitz on $\mathfrak{D}$, then $d_H(\Box\mathbf{f}(X), F(X)) \leq w(F(X))$ for any $X \in \mathfrak{D}$.*

Proof. Since $\mathbf{f}(X) \subset \Box\mathbf{f}(X) \subset F(X)$, this follows from [127, 1.7.3]. □

Define the typical inclusion functions for addition and multiplication: let the functions $(+, \mathbb{IR}_\emptyset^2, \mathbb{IR}_\emptyset)$ and $(\times, \mathbb{IR}_\emptyset^2, \mathbb{IR}_\emptyset)$ be defined by $+(X, Y) \equiv [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$ and $\times(X, Y) \equiv [\min(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\,\overline{y}), \max(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\,\overline{y})]$ and recall our convention[1] that any operation involving the empty interval results in an empty interval, i.e., $+(X, \emptyset) = +(\emptyset, X) = \emptyset$ or $\times(X, \emptyset) = \times(\emptyset, X) = \emptyset$ for any $X \in \mathbb{IR}_\emptyset$.

Assumption 3.1. Assume that for every $(u, B, \mathbb{R}) \in \mathcal{L}$, an interval extension $(u, \mathbb{I}_\emptyset B, \mathbb{IR}_\emptyset)$ is known. Furthermore, assume that this interval extension is inclusion monotonic on $\mathbb{I}_\emptyset B$.

Assumption 3.2. Assume that for every $(u, B, \mathbb{R}) \in \mathcal{L}$, the interval extension $(u, \mathbb{I}_\emptyset B, \mathbb{IR}_\emptyset)$ is locally Lipschitz on $\mathbb{I}_\emptyset B$.

Define the typical inclusion functions for the negative and reciprocal: let the functions $(-, \mathbb{IR}_\emptyset, \mathbb{IR}_\emptyset)$ and $(\frac{1}{(\cdot)}, \mathbb{I}_\emptyset(\mathbb{R} - \{0\}), \mathbb{IR}_\emptyset)$ be defined by $-(X) \equiv [-\overline{x}, -\underline{x}]$ and by $\frac{1}{X} \equiv [\frac{1}{\overline{x}}, \frac{1}{\underline{x}}]$, respectively. Note that these definitions satisfy Assumptions 3.1 and 3.2. Below, for convenience, we will write $X - Y \equiv X + (-Y)$ for some intervals X, Y.

3.2.1 Natural interval extensions

Suppose that Assumption 3.1 holds and that $(\mathcal{S}, \pi_o)$ is a $\mathcal{L}$-computational sequence. Then, to any element (o_k, π_k) of $\mathcal{S}$ a corresponding $(o_k, \mathbb{I}_\emptyset B_k, \mathbb{IR}_\emptyset)$ exists. Also, the functions $(\pi_k, \mathbb{IR}_\emptyset^{k-1}, \mathbb{IR}_\emptyset)$ or $(\pi_k, \mathbb{IR}_\emptyset^{k-1}, \mathbb{IR}_\emptyset^2)$ with $\pi_k(V) = (V_i)$ or $\pi_k(V) = (V_i, V_j)$ extend $(\pi_k, \mathbb{R}^{k-1}, \mathbb{R})$ or $(\pi_k, \mathbb{R}^{k-1}, \mathbb{R}^2)$ naturally.

Definition 3.12. For every $\mathcal{L}$-computational sequence $(\mathcal{S}, \pi_o)$ with n_i inputs and n_o outputs, define the *sequence of inclusion factors* $\{(V_k, \mathfrak{D}_k, \mathbb{IR}_\emptyset)\}_{k=1}^{n_f}$ where

1. for all $k = 1, \ldots, n_i$, $\mathfrak{D}_k = \mathbb{IR}_\emptyset^{n_i}$ and $V_k(X) = X_k$, $\forall X \in \mathfrak{D}_k$,
2. for all $k = n_i + 1, \ldots, n_f$, $\mathfrak{D}_k = \{X \in \mathfrak{D}_{k-1} : \pi_k(V_1(X), \ldots, V_{k-1}(X)) \in \mathbb{I}_\emptyset B_k\}$ and $V_k(X) = o_k(\pi_k(V_1(X), \ldots, V_{k-1}(X)))$, $\forall X \in \mathfrak{D}_k$.

The *natural interval extension* of $(\mathcal{S}, \pi_o)$ is the function $(F_\mathcal{S}, \mathfrak{D}_\mathcal{S}, \mathbb{IR}_\emptyset^{n_o})$ defined by $\mathfrak{D}_\mathcal{S} \equiv \mathfrak{D}_{n_f}$ and $F_\mathcal{S}(X) = \pi_o(V_1(X), \ldots, V_{n_f}(X))$, $\forall X \in \mathfrak{D}_\mathcal{S}$.

[1] Hereafter, we will not make this distinction explicitly in expressions. Rather it is always assumed tacitly.

Theorem 3.3. *Let $(\mathcal{S}, \pi_o)$ be a $\mathcal{L}$-computational sequence with associated natural function $(\mathbf{f}_\mathcal{S}, D_\mathcal{S}, \mathbb{R}^{n_o})$. The natural interval extension $(F_\mathcal{S}, \mathfrak{D}_\mathcal{S}, \mathbb{IR}^{n_o}_\emptyset)$ is an inclusion monotonic interval extension of $(\mathbf{f}_\mathcal{S}, D_\mathcal{S}, \mathbb{R}^{n_o})$ on $\mathfrak{D}_\mathcal{S}$ and an inclusion function for $\mathbf{f}_\mathcal{S}$ on $\mathfrak{D}_\mathcal{S}$. In particular, each inclusion factor V_k of $(\mathcal{S}, \pi_o)$ is an inclusion monotonic interval extension of v_k on $\mathfrak{D}_\mathcal{S}$ for all $k = 1, \ldots, n_f$.*

Proof. See [155, Theorem 2.3.11] together with Theorem 3.1. □

Definition 3.13. Let $\mathbf{f} : D \subset \mathbb{R}^n \to \mathbb{R}^m$ be a $\mathcal{L}$-factorable function. Then, for any $\mathcal{L}$-computation sequence describing $\mathbf{f}$, the natural interval extension $(F_\mathcal{S}, \mathfrak{D}_\mathcal{S}, \mathbb{IR}^m)$ is called a natural interval extension of $\mathbf{f}$. It will be denoted as $(F, \mathfrak{D}, \mathbb{IR}^m)$.

Remark 3.1. Refer to [127, Sec. 2.1] or [155, Sec. 2.5.5] for conditions that imply that the natural interval extension F of a $\mathcal{L}$-factorable function $\mathbf{f}$ is locally Lipschitz on $\mathfrak{D}$. If Assumption 3.2 holds, this is the case as shown in [155, Theorem 2.5.30]. In this case, Theorem 3.2 and Lemma 3.1 indicate that the overestimation error of the range, $d_H(\square\mathbf{f}(X), F(X))$, is linear in $w(X)$ for any $X \in \mathfrak{D}_\mathcal{S}$. In other words, the natural interval extension of $\mathbf{f}$ has linear convergence order since $\min_{\mathbf{x}\in X} f_i(\mathbf{x}) - \min_{\mathbf{x}\in X} F_i(X) = \min_{\mathbf{x}\in X} f_i(\mathbf{x}) - \underline{f}_i(X) \leq d_H(f_i(X), F_i(X))$. It is also possible to calculate a pre-factor K in this case, cf. [127, Sec. 2.1].

Since the construction of natural interval extensions can be easily automated, different software packages have been developed. One possible approach to implementing a library for these computations is to use operator overloading in a object-oriented programming language such as `C++`. Readily available implementations include `PROFIL/BIAS` [99] and `Boost` [120]. Note that the cost of evaluating a natural interval extension is only a small multiple of the cost of evaluating the real-valued function.

Two of the major downsides of interval arithmetic are the *dependency problem* and the *wrapping effect* [91, 122]. The dependency problem refers to the fact that interval arithmetic is a memory-less computation. This can be illustrated with a simple example. Consider factor v_7 in Example 3.1 that depends on the factors v_4 and v_6. Since both v_4 and v_6 depend on v_2, they cannot vary independently of each other. However, interval arithmetic presumes that this is indeed true and thus introduces overestimation on the range of v_7 when V_7 is computed. The wrapping effect refers to the overestimation due to the poor approximation of non-rectangular shapes with intervals. Since intervals are the only objects available to describe more complex geometries within the realm of interval analysis, considerable overestimation can be introduced this way. While extensions of interval arithmetic have been introduced, e.g., affine arithmetic [161], these are not further considered in this body of work.

3.2.2 Centered forms

Since the amount by which natural interval extensions overestimate the range of the function is linear in the width of the interval over which the interval extension is constructed, see Remark 3.1, others have constructed and investigated interval methods with quadratic

convergence order, sometimes referred to as the *quadratic approximation property* in the literature [e.g., 138]. Centered forms were first suggested by Moore [122, p. 44f], also see [18, 137]. These methods can be motivated by the mean value theorem in the real numbers [145].

Theorem 3.4. *Suppose $D \subset \mathbb{R}^n$ is open and $f : D \to \mathbb{R}$ is differentiable on D. Then, for any $\mathbf{x}, \mathbf{c} \in D$ there exists a $\lambda \in [0,1]$ so that $f(\mathbf{x}) = f(\mathbf{c}) + \nabla f(\boldsymbol{\xi})(\mathbf{x} - \mathbf{c})$ where $\boldsymbol{\xi} = \lambda\mathbf{x} + (1-\lambda)\mathbf{c}$.*

This result motivates the following interval extension of f.

Definition 3.14. Suppose $D \subset \mathbb{R}^n$, $f : D \to \mathbb{R}$ and let $\mathfrak{D}$ be an interval extension of D. Assume that for each $X \in \mathfrak{D}$ and some fixed $\mathbf{c} \in X$ there exists $S(X, \mathbf{c}) \in \mathbb{IR}^n$ so that for all $\mathbf{x} \in X$ there exists a $\mathbf{s}(\mathbf{x}) \in S(X, \mathbf{c})$ such that $f(\mathbf{x}) = f(\mathbf{c}) + \mathbf{s}(\mathbf{x})^{\mathrm{T}}(\mathbf{x} - \mathbf{c})$ holds. Then, $F_c : \mathfrak{D} \times D \to \mathbb{R}$ given by $F_c(X, \mathbf{c}) = f(\mathbf{c}) + S(X, \mathbf{c})^{\mathrm{T}}(X - \mathbf{c})$ is called a *centered form* of f.

The following result establishes that the centered form is indeed an interval extension and yields a first convergence order result.

Theorem 3.5. *Suppose $D \subset \mathbb{R}^n$ is open and $f : D \to \mathbb{R}$ is differentiable on D. Then, $F_c(\cdot, \mathbf{c})$ is an inclusion function of f on $\mathfrak{D}$ for any valid $\mathbf{c}$. Furthermore, it holds that*

$$d_H(f(X), F_c(X, \mathbf{c})) \leq \sum_{i=1}^{n} w(S_i(X, \mathbf{c}))|X_i - c_i|, \forall X \in \mathfrak{D}, \mathbf{c} \in X.$$

Proof. See [127, Theorem 2.3.3]. □

One method to obtain a suitable S is to use the mean value theorem directly. Assume that D is open and that f is differentiable on D. Suppose ∇F is an interval extension of ∇f on $\mathbb{I}D$. Then, S can be replaced by ∇F. The obtained interval extension of f is called a *(generalized) mean value form*. Under certain assumptions, quadratic convergence can be established [104]. Hence, the mean value form possesses the desired property to mitigate the cluster effect.

Theorem 3.6. *Assume that D is open and that f is differentiable on D. Consider the mean value form F_c of f. Suppose that ∇F is Lipschitz continuous on $\mathbb{I}D$ such that there exists $L > 0$ with $w(\nabla F(X)) \leq L w(X)$. Then, $\beta = 2$ and $K = nL$.*

Proof. As shown in [127, 2.3.3], it holds that

$$\min_{\mathbf{x} \in X} f(\mathbf{x}) - \underline{F_c}(X, \mathbf{c}) \leq \sum_i w(\nabla F_i(X))|X_i - c_i| \leq nL w(X)^2, \forall X \in \mathbb{I}D, \mathbf{c} \in X. \tag{3.1}$$

Thus, $\beta = 2$ and an estimate for the convergence order pre-factor is given by $K = nL$. □

Neumaier [127] noted that a bound on $\min_{\mathbf{x} \in X} f(\mathbf{x}) - \underline{F_c}(X, \mathbf{c})$ can be obtained immediately *a posteriori* from Eq. 3.1, e.g., concurrent to the execution of a branch-and-bound algorithm.

Alternative methods have proposed to obtain S. For example, *slope forms* directly use the definition of a centered form to construct S [103, 127]. Similar to the natural interval extension, slopes can be constructed step-by-step for each element of the $\mathcal{L}$-computational sequence. In addition to the rules for the elementary binary arithmetic operations, Ratz [139] describes how to carry out the necessary calculations for some univariate functions. Hence, it is possible to automate the evaluation of a slope form so that an inclusion function with quadratic convergence o rder is available. In practice, it is best to use cheaper natural interval extensions on large host sets since the overestimation introduced using centered forms can be considerable on large host sets. Only on small sets does the quadratic convergence order provide a benefit that outweighs the additional computational expense.

3.3 McCormick analysis

Let $D \subset \mathbb{R}^n$ be convex. A vector-valued function $\mathbf{g} : D \to \mathbb{R}^m$ is *convex* on D if each component is convex on D. Similarly, it is called *concave* on D if each component is concave on D. For any set A, let $\mathbb{P}(A)$ denote the *power set*, or set of all subsets, including the empty set, of A.

Definition 3.15. Let $D \subset \mathbb{R}^n$ be a convex set and $\mathbf{f} : D \to \mathbb{P}(\mathbb{R}^m)$. A function $\check{\mathbf{f}} : D \to \mathbb{R}^m$ is a *convex relaxation*, or *convex underestimator*, of $\mathbf{f}$ on D if $\check{\mathbf{f}}$ is convex on D and $\check{f}_i(\mathbf{x}) \leq \inf\{f_i(\mathbf{x})\}$, $\forall \mathbf{x} \in D$ and $i = 1, \ldots, m$. A convex relaxation $\mathbf{g} : D \to \mathbb{R}^m$ is called the *convex envelope* of $\mathbf{f}$ on D if $g_i(\mathbf{x}) \geq \check{f}_i(\mathbf{x})$ for all convex relaxations of $\mathbf{f}$, $\forall \mathbf{x} \in D$ and $i = 1, \ldots, m$. A function $\hat{\mathbf{f}} : D \to \mathbb{R}^m$ is a *concave relaxation*, or *concave overestimator*, of $\mathbf{f}$ on D if $\hat{\mathbf{f}}$ is concave on D and $\hat{f}_i(\mathbf{x}) \geq \sup\{f_i(\mathbf{x})\}$, $\forall \mathbf{x} \in D$ and $i = 1, \ldots, m$. A concave relaxation $\mathbf{g} : D \to \mathbb{R}^m$ is called the *concave envelope* of $\mathbf{f}$ on D if $g_i(\mathbf{x}) \leq \hat{f}_i(\mathbf{x})$ for all concave relaxations of $\mathbf{f}$, $\forall \mathbf{x} \in D$ and $i = 1, \ldots, m$.

Remark 3.2. Definition 3.15 allows that $\mathbf{f}(\mathbf{x}) = \emptyset$ for some $\mathbf{x} \in D$. In this case, the inequality defining a relaxation will hold for any function. However, the convexity and concavity requirement must still be met by $\check{\mathbf{f}}$ and $\hat{\mathbf{f}}$, respectively, and this requirement constrains the set of functions that satisfy the definition, as exemplified in Figure 4.7.

The following notation was introduced in [155]. While it differs from the previously used notation for McCormick relaxations, it is more compact and very useful for the proposed operations on computational sequences, and it also makes the relationship with interval arithmetic more apparent. In the latter, information is passed from one operation in the sequence of factors to the next in the forms of intervals. McCormick's procedure to construct relaxations [118], on the other hand, requires an interval X and a point $\mathbf{x} \in X$ as input and returns three values: an interval $V_k(X)$, which encloses the image of X under v_k, and two additional values $\check{v}_k(X, \mathbf{x})$ and $\hat{v}_k(X, \mathbf{x})$, which represent the value of the convex and concave relaxation of v_k on X evaluated at $\mathbf{x}$. After a recent generalization, one can also consider mappings that take an interval and two relaxation values as input and return an interval and two relaxation values; these are called generalized McCormick

relaxations [156]. One advantage of this generalization is that it yields mappings with conformable inputs and outputs, which are hence composable.

Definition 3.16. Let $\mathbb{MR}^n \equiv \{(Z^B, Z^C) \in \mathbb{IR}^n \times \mathbb{IR}^n : Z^B \cap Z^C \neq \emptyset\}$. Elements of $\mathbb{MR}^n$ are denoted by script capitals, $\mathcal{Z} \in \mathbb{MR}^n$. For any $\mathcal{Z} \in \mathbb{MR}^n$, the notations $Z^B, Z^C \in \mathbb{IR}^n$ and $(\underline{\mathbf{z}}, \overline{\mathbf{z}}, \check{\mathbf{z}}, \hat{\mathbf{z}}) \in \mathbb{R}^n$ will be commonly used to denote the intervals and vectors satisfying $\mathcal{Z} = (Z^B, Z^C) = ([\underline{\mathbf{z}}, \overline{\mathbf{z}}], [\check{\mathbf{z}}, \hat{\mathbf{z}}])$. For any $D \subset \mathbb{R}^n$, let $\mathbb{M}D$ denote the set $\{\mathcal{Z} \in \mathbb{MR}^n : Z^B \subset D\}$.

In this thesis, it is also necessary to consider unbounded and empty McCormick objects. Analogously to Definition 3.7, define the sets $\mathbb{MR}^n_\emptyset \equiv \{(Z^B, Z^C) \in \mathbb{IR}^n_\emptyset \times \mathbb{IR}^n_\emptyset : Z^B \cap Z^C \neq \emptyset \vee Z^C = \emptyset\}$ and $\bar{\mathbb{M}}\mathbb{R}^n \equiv \{(Z^B, Z^C) \in \bar{\mathbb{I}}\mathbb{R}^n \times \bar{\mathbb{I}}\mathbb{R}^n : Z^B \cap Z^C \neq \emptyset \vee Z^C = \emptyset\}$, which are extensions of $\mathbb{MR}^n$. Also, define $\mathbb{M}_\emptyset D$ and $\bar{\mathbb{M}}D$ for any $D \in \mathbb{R}^n$ analogous to $\mathbb{I}_\emptyset D$ and $\bar{\mathbb{I}}D$. Introduce $\mathrm{Enc} : \bar{\mathbb{M}}\mathbb{R}^n \to \bar{\mathbb{I}}\mathbb{R}^n$ defined by $\mathrm{Enc}(\mathcal{Z}) = Z^B \cap Z^C$ for all $\mathcal{Z} \in \bar{\mathbb{M}}\mathbb{R}^n$. This function is necessary since for $\mathbf{z} \in \mathbb{R}^n_\emptyset$, $\mathbf{z} \in \mathcal{Z}$ is not well-defined whereas $\mathbf{z} \in \mathrm{Enc}(\mathcal{Z})$ is.

Next, we formalize McCormick's technique by defining operations on $\mathbb{MR}^n_\emptyset$. We introduce the concept of a *relaxation function*, which is analogous to the notion of an inclusion function in interval analysis, and is the fundamental object that we want to compute for a given real-valued function. Then, we show how relaxation functions can be obtained through a simpler construction, just as inclusion functions can be constructed from inclusion monotonic interval extensions. First, however, some preliminary concepts are necessary.

Definition 3.17. Let $\mathcal{X}, \mathcal{Y} \in \bar{\mathbb{M}}\mathbb{R}^n$. $\mathcal{X}$ and $\mathcal{Y}$ are *coherent* if $X^B = Y^B$. A set $\mathcal{D} \subset \bar{\mathbb{M}}\mathbb{R}^n$ is coherent if every pair of elements is coherent. A set $\mathcal{D} \subset \bar{\mathbb{M}}\mathbb{R}^n$ is *closed under coherence* if, for any coherent $\mathcal{X}, \mathcal{Y} \in \bar{\mathbb{M}}\mathbb{R}^n$, $\mathcal{X} \in \mathcal{D}$ implies $\mathcal{Y} \in \mathcal{D}$.

For any coherent $\mathcal{X}_1, \mathcal{X}_2 \in \bar{\mathbb{M}}\mathbb{R}^n$ with common interval part Q and for all $\lambda \in [0,1]$, define

$$\mathrm{Conv}(\lambda, \mathcal{X}_1, \mathcal{X}_2) \equiv (Q, \lambda X_1^C + (1-\lambda) X_2^C)$$

where the rules of interval arithmetic are used to evaluate $\lambda X_1^C + (1-\lambda) X_2^C$. For any $\mathcal{X}_1, \mathcal{X}_2 \in \bar{\mathbb{M}}\mathbb{R}^n$, the inclusion $\mathcal{X}_1 \subset \mathcal{X}_2$ holds iff $X_1^B \subset X_2^B$ and $X_1^C \subset X_2^C$. Likewise, $\mathcal{X}_1 \supset \mathcal{X}_2$ iff $\mathcal{X}_2 \subset \mathcal{X}_1$. Also, define $\mathcal{X}_1 \cap \mathcal{X}_2 \equiv (X_1^B \cap X_2^B, X_1^C \cap X_2^C)$.

Definition 3.18. Suppose $\mathcal{D} \subset \bar{\mathbb{M}}\mathbb{R}^n$ is closed under coherence. A function $\mathcal{F} : \mathcal{D} \to \bar{\mathbb{M}}\mathbb{R}^m$ is *coherently concave* on $\mathcal{D}$ if for every coherent $\mathcal{X}_1, \mathcal{X}_2 \in \mathcal{D}$, i.e., $Q = X_1^B = X_2^B$, $\mathcal{F}(\mathcal{X}_1)$ and $\mathcal{F}(\mathcal{X}_2)$ are coherent, and $\mathcal{F}(\mathrm{Conv}(\lambda, \mathcal{X}_1, \mathcal{X}_2)) \supset \mathrm{Conv}(\lambda, \mathcal{F}(\mathcal{X}_1), \mathcal{F}(\mathcal{X}_2))$ for every $\lambda \in [0,1]$.

Definition 3.19. Let $\mathbf{f} : D \subset \mathbb{R}^n \to \mathbb{R}^m_\emptyset$. A mapping $\mathcal{F} : \mathcal{D} \subset \bar{\mathbb{M}}D \to \bar{\mathbb{M}}\mathbb{R}^m$ is a *relaxation function* for $\mathbf{f}$ on $\mathcal{D}$ if $\mathcal{D}$ is closed under coherence, $\mathcal{F}$ is coherently concave on $\mathcal{D}$, and $\mathbf{f}(\mathbf{x}) \in \mathrm{Enc}(\mathcal{F}(\mathcal{X}))$ is satisfied for every $\mathcal{X} \in \mathcal{D}$ and $\mathbf{x} \in \mathrm{Enc}(\mathcal{X})$.

Remark 3.3. Note that Definition 3.18 is a generalization of convexity and concavity, and Definition 3.19 is a generalization of the notion of convex and concave relaxations. Convex and concave relaxations of $\mathbf{f}$ can be recovered from a relaxation function of $\mathbf{f}$ as follows. Let

$X \in \mathbb{I}D$ so that there exists $\mathcal{Y} \in \mathcal{D}$ with $X = Y^B$. Define the functions $\mathcal{U}, \mathcal{O} : X \to \mathbb{R}^m_\emptyset$ for all $\mathbf{x} \in X$ by $([\underline{\mathbf{f}}, \overline{\mathbf{f}}], [\mathcal{U}(\mathbf{x}), \mathcal{O}(\mathbf{x})]) \equiv \mathcal{F}((X, [\mathbf{x}, \mathbf{x}]))$. Then, $\mathcal{U}$ and $\mathcal{O}$ are convex and concave relaxations of $\mathbf{f}$ on X, respectively, as shown in [155, Lemma 2.4.11].

Similar to the notion of an inclusion function in interval analysis, the definition of a relaxation function provides a construct with properties relevant in global optimization. And again, the enclosure property of the relaxation function can be obtained with a simpler construction as shown below.

Definition 3.20. Let $D \subset \mathbb{R}^n$. A set $\mathcal{D} \subset \mathbb{M}\mathbb{R}^n$ is a *McCormick extension* of D if $\mathcal{D} \subset \mathbb{M}D$ and every $\mathbf{x} \in D$ satisfies $([\mathbf{x}, \mathbf{x}], [\mathbf{x}, \mathbf{x}]) \in \mathcal{D}$. Let $\mathbf{f} : D \to \mathbb{R}^m_\emptyset$. A function $\mathcal{F} : \mathcal{D} \to \mathbb{M}\mathbb{R}^m$ is a *McCormick extension* of $\mathbf{f}$ if $\mathcal{D}$ is a McCormick extension of D and $\mathcal{F}([\mathbf{x}, \mathbf{x}], [\mathbf{x}, \mathbf{x}]) = ([\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x})], [\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x})])$, $\forall \mathbf{x} \in D$.

Definition 3.21. Let $\mathcal{F} : \mathcal{D} \subset \mathbb{M}\mathbb{R}^n \to \mathbb{M}\mathbb{R}^m$. $\mathcal{F}$ is *inclusion monotonic* on $\mathcal{D}$ if $\mathcal{X}_1 \subset \mathcal{X}_2$ implies that $\mathcal{F}(\mathcal{X}_1) \subset \mathcal{F}(\mathcal{X}_2)$ for all $\mathcal{X}_1, \mathcal{X}_2 \in \mathcal{D}$.

Theorem 3.7. *Let $\mathbf{f} : D \subset \mathbb{R}^n \to \mathbb{R}^m_\emptyset$ and let $\mathcal{F} : \mathcal{D} \to \mathbb{M}\mathbb{R}^m$ be a McCormick extension of $\mathbf{f}$. If $\mathcal{F}$ is inclusion monotonic on $\mathcal{D}$, then every $\mathcal{X} \in \mathcal{D}$ satisfies $\mathbf{f}(\mathbf{x}) \in \mathrm{Enc}(\mathcal{F}(\mathcal{X}))$ for all $\mathbf{x} \in \mathrm{Enc}(\mathcal{X})$.*

Proof. See [155, Theorem 2.4.14]. □

We conclude that an inclusion monotonic McCormick extension that is also coherently concave is a relaxation function. Hence, it suffices to derive an inclusion monotonic, coherently concave McCormick extension. As shown in [155, Lemmas 2.4.15, 2.4.17] the composition of inclusion monotonic, coherently concave McCormick extensions yields an inclusion monotonic, coherently concave McCormick extension. This motivates the derivations of inclusion monotonic, coherently concave McCormick extensions of the basic operations below.

Define the following relaxation functions for addition and multiplication: let the functions $(+, \mathbb{M}\mathbb{R}^2_\emptyset, \mathbb{M}\mathbb{R}_\emptyset)$ and $(\times, \mathbb{M}\mathbb{R}^2_\emptyset, \mathbb{M}\mathbb{R}_\emptyset)$ be given by $+(\mathcal{X}, \mathcal{Y}) = (X^B + Y^B, X^C + Y^C)$ and $\times(\mathcal{X}, \mathcal{Y}) = (X^B Y^B, [\check{z}, \hat{z}])$ where

$$\check{z} = \max\left(\left(\underline{y}X^C + \underline{x}Y^C - \underline{x}\,\underline{y}\right)^L, \left(\overline{y}X^C + \overline{x}Y^C - \overline{x}\,\overline{y}\right)^L, (X^B Y^B)^L\right)$$

and

$$\hat{z} = \min\left(\left(\underline{y}X^C + \overline{x}Y^C - \overline{x}\underline{y}\right)^U, \left(\overline{y}X^C + \underline{x}Y^C - \underline{x}\overline{y}\right)^U, (X^B Y^B)^U\right).$$

Note that this definition of multiplication ensures that $[\check{z}, \hat{z}] \subset X^B Y^B$[155, Theorems 2.4.22, 2.4.23]. Furthermore, the standard rules for addition and multiplication of McCormick relaxations are implied by these definitions, see [155, p. 69f], with the addition of the intersection with the bounds from interval arithmetic in the case of the multiplication rule. These functions are indeed relaxation functions and also inclusion monotonic as shown in [155, Section 2.4.2].

The following assumption is needed to construct relaxation functions for the elements of $\mathcal{L}$. For many univariate functions, objects satisfying these assumptions are known and readily available [155, Section 2.8].

Assumption 3.3. Assume that for every $(u, B, \mathbb{R}) \in \mathcal{L}$, functions $\check{u}, \hat{u} : \tilde{B} \to \mathbb{R}$ where $\tilde{B} \equiv \{(X, x) \in \mathbb{I}B \times B : x \in X\}$ and $x^{\min}, x^{\max} : \mathbb{I}B \to \mathbb{R}$ are known such that $\check{u}(X, \cdot)$ and $\hat{u}(X, \cdot)$ are convex and concave relaxations of u on $X \in \mathbb{I}B$, respectively, and $x^{\min}(X)$ and $x^{\max}(X)$ are a minimum of $\check{u}(X, \cdot)$ and a maximum of $\hat{u}(X, \cdot)$ on X, respectively. Furthermore, assume that for any $X_1, X_2 \in \mathbb{I}B$ with $X_1 \subset X_2$, $\check{u}(X_1, x) \geq \check{u}(X_2, x)$ and $\hat{u}(X_1, x) \leq \hat{u}(X_2, x)$ for all $x \in X_1$ and that $\check{u}([x, x], x) = \hat{u}([x, x], x)$ for all $x \in B$.

Let $\mathrm{mid} : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ return the middle value of its arguments. It can be shown [cf. 155, p. 76f] that a relaxation function of $(u, B, R) \in \mathcal{L}$ is given by $(u, \mathbb{M}B, \mathbb{M}\mathbb{R})$ with

$$u(\mathcal{X}) = \left(u(X^B), \left[\check{u}(X^B, \mathrm{mid}(\check{x}, \hat{x}, x^{\min}(X^B))), \hat{u}(X^B, \mathrm{mid}(\check{x}, \hat{x}, x^{\max}(X^B)))\right] \cap u(X^B)\right). \tag{3.2}$$

Note that if the convex and concave envelopes of u are known and used, then the intersection with $u(X^B)$ in (3.2) is redundant.

Assumption 3.4. Assume that for every $(u, B, \mathbb{R}) \in \mathcal{L}$, the relaxation function $(u, \mathbb{M}B, \mathbb{M}\mathbb{R})$ is locally Lipschitz on $\mathbb{M}B$.

As reported in [156, Supplementary material], Assumptions 3.3 and 3.4 are satisfied for the negative function with the definitions $B = \mathbb{R}$, $x^{\min}(X) = \overline{x}$, $x^{\max}(X) = \underline{x}$, $\check{u}(X, x) = -x$ and $\hat{u}(X, x) = -x$ and for the reciprocal function with the definitions $B = \mathbb{R} - \{0\}$, $x^{\min}(X) = \overline{x}$, $x^{\max}(X) = \underline{x}$

$$\check{u}(X, x) = \begin{cases} \frac{1}{x} & \text{if } \underline{x} > 0, \\ \frac{1}{\underline{x}} - \frac{1}{\underline{x}\overline{x}}(x - \underline{x}) & \text{if } \overline{x} < 0, \end{cases} \quad \text{and} \quad \hat{u}(X, x) = \begin{cases} \frac{1}{\underline{x}} - \frac{1}{\underline{x}\overline{x}}(x - \underline{x}) & \text{if } \underline{x} > 0, \\ \frac{1}{x} & \text{if } \overline{x} < 0. \end{cases}$$

Below, for convenience, we will write $\mathcal{X} - \mathcal{Y} \equiv \mathcal{X} + (-\mathcal{Y})$ for $\mathcal{X}, \mathcal{Y} \in \bar{\mathbb{M}}\mathbb{R}$.

3.3.1 Natural McCormick extensions

Just as there is a natural way in which to extend the real-valued calculations on the sequence of factors of a $\mathcal{L}$-computational sequence to interval arithmetic to obtain the natural interval extension, this extension is also possible for the more complex McCormick objects [155, 156].

Suppose that Assumption 3.3 holds and that $(\mathcal{S}, \pi_o)$ is a $\mathcal{L}$-computational sequence. Then, for any element (o_k, π_k) of $\mathcal{S}$, the preceding developments provide an inclusion monotonic McCormick extension $(o_k, \mathbb{M}_\emptyset B_k, \mathbb{M}\mathbb{R}_\emptyset)$ exists. Also, the functions $(\pi_k, \mathbb{M}\mathbb{R}_\emptyset^{k-1}, \mathbb{M}\mathbb{R}_\emptyset)$ or $(\pi_k, \mathbb{M}\mathbb{R}_\emptyset^{k-1}, \mathbb{M}\mathbb{R}_\emptyset^2)$ with $\pi_k(\mathcal{V}) = (\mathcal{V}_i)$ or $\pi_k(\mathcal{V}) = (\mathcal{V}_i, \mathcal{V}_j)$ extend $(\pi_k, \mathbb{R}^{k-1}, \mathbb{R})$ or $(\pi_k, \mathbb{R}^{k-1}, \mathbb{R}^2)$ naturally.

Definition 3.22. For every $\mathcal{L}$-computational sequence $(\mathcal{S}, \pi_o)$ with n_i inputs and n_o outputs, define the *sequence of relaxation factors* $\{(\mathcal{V}_k, \mathcal{D}_k, \mathbb{M}\mathbb{R}_\emptyset)\}_{k=1}^{n_f}$ where

1. for all $k = 1, \ldots, n_i$, $\mathcal{D}_k = \mathbb{MR}_{\emptyset}^{n_i}$ and $\mathcal{V}_k(\mathcal{X}) = \mathcal{X}_k$, $\forall \mathcal{X} \in \mathcal{D}_k$,
2. for all $k = n_i + 1, \ldots, n_f$, $\mathcal{D}_k = \{\mathcal{X} \in \mathcal{D}_{k-1} : \pi_k(\mathcal{V}_1(\mathcal{X}), \ldots, \mathcal{V}_{k-1}(\mathcal{X})) \in \mathbb{M}_{\emptyset} B_k\}$ and $\mathcal{V}_k(\mathcal{X}) = o_k(\pi_k(\mathcal{V}_1(\mathcal{X}), \ldots, \mathcal{V}_{k-1}(\mathcal{X})))$, $\forall \mathcal{X} \in \mathcal{D}_k$.

The *natural McCormick extension* of $(\mathcal{S}, \pi_o)$ is the function $(\mathcal{F}_{\mathcal{S}}, \mathcal{D}_{\mathcal{S}}, \mathbb{MR}^{n_o})$ defined by $\mathcal{D}_{\mathcal{S}} \equiv \mathcal{D}_{n_f}$ and $\mathcal{F}_{\mathcal{S}}(\mathcal{X}) = \pi_o(\mathcal{V}_1(\mathcal{X}), \ldots, \mathcal{V}_{n_f}(\mathcal{X}))$, $\forall \mathcal{X} \in \mathcal{D}_{\mathcal{S}}$.

Theorem 3.8. *Let $(\mathcal{S}, \pi_o)$ be a $\mathcal{L}$-computational sequence with associated natural function $(\mathbf{f}_{\mathcal{S}}, D_{\mathcal{S}}, \mathbb{R}^{n_o})$. The natural McCormick extension $(\mathcal{F}_{\mathcal{S}}, \mathcal{D}_{\mathcal{S}}, \mathbb{MR}_{\emptyset}^{n_o})$ is a McCormick extension of $(\mathbf{f}_{\mathcal{S}}, D_{\mathcal{S}}, \mathbb{R}^{n_o})$ and coherently concave and inclusion monotonic on $\mathcal{D}_{\mathcal{S}}$. Thus, it is a relaxation function for $\mathbf{f}_{\mathcal{S}}$ on $\mathcal{D}_{\mathcal{S}}$. In particular, each relaxation factor $\mathcal{V}_k$ of $(\mathcal{S}, \pi_o)$ is a inclusion monotonic, coherently concave McCormick extension of v_k on $\mathfrak{D}_{\mathcal{S}}$ for all $k = 1, \ldots, n_f$.*

Proof. Follows from [155, Theorem 2.4.32] together with Theorem 3.7. □

Definition 3.23. Let $\mathbf{f} : D \subset \mathbb{R}^n \to \mathbb{R}^m$ be $\mathcal{L}$-factorable. Then, for any $\mathcal{L}$-computational sequence describing $\mathbf{f}$, the natural McCormick extension $(\mathcal{F}_{\mathcal{S}}, \mathcal{D}_{\mathcal{S}}, \mathbb{MR}^m)$ is called a natural McCormick extension of $\mathbf{f}$.

3.3.2 Standard McCormick relaxations

Definition 3.24. Let $\mathbf{f} : D \subset \mathbb{R}^n \to \mathbb{R}^m$ be $\mathcal{L}$-factorable and let $\mathcal{F} : \mathcal{D} \subset \mathbb{MR}^n \to \mathbb{MR}^m$ be a natural McCormick extension of $\mathbf{f}$. For any $X \in \mathbb{I}D$ that is represented in $\mathcal{D}$, define $\check{\mathbf{f}}, \hat{\mathbf{f}} : X \to \mathbb{R}^m$ by

$$\check{\mathbf{f}}(\mathbf{x}) \equiv \check{\mathbf{f}}((X, [\mathbf{x}, \mathbf{x}])) \text{ and } \hat{\mathbf{f}}(\mathbf{x}) \equiv \hat{\mathbf{f}}((X, [\mathbf{x}, \mathbf{x}])).$$

These functions are called *standard McCormick relaxations of* $\mathbf{f}$ *on* X.

Standard McCormick relaxations have been shown to converge quadratically [34] in the following sense when convex and concave envelopes are used to construct relaxations of univariate functions.

Theorem 3.9. *Suppose that $D \subset \mathbb{R}^n$ is open. Let $\mathbf{f} : D \to \mathbb{R}^m$ be $\mathcal{L}$-factorable and let $\mathcal{F} : \mathcal{D} \subset \mathbb{MR}^n \to \mathbb{MR}^m$ be a natural McCormick extension of $\mathbf{f}$. Assume that $\mathbf{f}$ is twice differentiable on D. Suppose that for each $(u, B, \mathbb{R}) \in \mathcal{L}$ convex and concave envelopes $\check{u}$ and $\hat{u}$ are used. Introduce $[\mathbf{f}_*(X), \mathbf{f}^*(X)] \equiv \cup_{\mathbf{x} \in X} [\check{\mathbf{f}}(\mathbf{x}), \hat{\mathbf{f}}(\mathbf{x})]$ for any $X \in \mathbb{I}D$. Then there exists a $K > 0$ so that*

$$d_H(\square \mathbf{f}(X), [\mathbf{f}_*(X), \mathbf{f}^*(X)]) \leq K w(X)^2, \ \forall X \in \mathbb{I}D.$$

Proof. Note that $f_{*,i} = \inf_{\mathbf{x} \in X} \check{f}_i(\mathbf{x})$ and $f_i^* = \sup_{\mathbf{x} \in X} \check{f}_i(\mathbf{x})$. The conclusion thus follows from the results in [34]. □

The automatic computation of natural McCormick relaxations of a factorable function has been described in [121]. A `C++` library, `MC++` [40], which is the successor of the `C++` library described in [121], is available to evaluate natural McCormick relaxations using operator overloading. Additionally, this library provides means to evaluate a member

of the subdifferential of f_i and $\hat{f}_i$, $i = 1, \ldots, n_o$ using ideas similar to the forward mode of automatic differentiation [70]. Others have presented similar libraries to compute subgradients of the relaxations using either the forward [48] or the reverse mode of automatic differentiation [21] that are based on source code transformation.

3.4 αBB relaxations

αBB relaxations [2, 5, 116] provide an alternative for constructing convex relaxations. The underlying idea is not as strongly intertwined with the concept of a $\mathcal{L}$-computational sequence and its natural function. For this reason, it will only briefly be mentioned and not further investigated in the remainder of the thesis.

Suppose $D \subset \mathbb{R}^n$ is open and $f : D \to \mathbb{R}$ is twice differentiable on D. In the most general case, αBB relaxations of f on $X \in \mathbb{I}D$ are defined as the function $f_\alpha : X \to \mathbb{R}$ given by

$$f_\alpha(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^{n} \alpha_i(X)(\underline{x}_i - x_i)(\overline{x}_i - x_i), \forall \mathbf{x} \in X$$

where $\alpha_i(X)$ are non-negative reals that are sufficiently large to guarantee convexity of f_α on X. Different methods have been proposed to calculate α_i [2, 116] and $\alpha_i(X)$ can be updated as X changes. When the method was first proposed, its quadratic convergence order was demonstrated [116]. The main focus here is to establish a bound on the convergence order pre-factor K, which can be obtained by the following result regardless of the method.

Theorem 3.10. *Consider αBB relaxations of f and suppose C is an interval. Let $\alpha \equiv \max_{i=1,\ldots,n} \alpha_i$ where α_i has been calculated on C. Then, $\beta = 2$ and $K \leq \frac{1}{4}\alpha n$.*

Proof. It is easy to see that, for any $X \in \mathbb{I}C$,

$$\begin{aligned}
\min_{\mathbf{x}\in X} f(\mathbf{x}) - \min_{\mathbf{x}\in X} f_\alpha(\mathbf{x}) &= \min_{\mathbf{x}\in X} f(\mathbf{x}) - \min_{\mathbf{x}\in X} \left(f(\mathbf{x}) + \sum_{i=1}^{n} \alpha_i(\underline{x}_i - x_i)(\overline{x}_i - x_i) \right) \\
&\leq \min_{\mathbf{x}\in X} f(\mathbf{x}) - \min_{\mathbf{x}\in X} f(\mathbf{x}) - \min_{\mathbf{x}\in X} \sum_{i=1}^{n} \alpha_i(\underline{x}_i - x_i)(\overline{x}_i - x_i) \\
&= \sum_{i=1}^{n} \alpha_i \left(\frac{w(X_i)}{2} \right)^2 \leq \frac{1}{4}\alpha n (w(X))^2.
\end{aligned}$$

Thus, it follows that $\beta = 2$ and that $K = \frac{1}{4}\alpha n$ is a conservative estimate of the pre-factor. □

In Section 2.2 it was remarked that $K \leq \frac{9\lambda_1}{4}$ must hold in order to prevent the exponential growth of N with n for a second-order relaxation. For αBB relaxations this condition translates to $\alpha \leq \frac{9\lambda_1}{n}$. Recall that $\lambda_1 > 0$ is the smallest eigenvalue of $\nabla^2 f(\mathbf{x}^*)$, not the smallest eigenvalue of $\nabla^2 f$ on C. Note also that Theorem 3.10 does not indicate whether

Method	Complexity	Convergence order
Natural interval extensions	$O(n_f)$	$Kw(X)$
Centered forms	$O(nn_f)$	$nKw(X)^2$
McCormick relaxations	$O(n_f)$	$Kw(X)^2$
αBB relaxations	$O(n^2 n_f)$	$\frac{1}{4}\alpha n w(X)^2$

Table 3.2: Comparison of the computational complexity and the convergence order of different bounding methods. For reference, the complexity of one evaluation of the natural function is $O(n_f)$.

αBB relaxations can achieve this criterion. Furthermore, the result assumes that α_i does not change with X.

Suppose now that we construct a new α on each interval visited. A note-worthy feature of αBB relaxations is that $\alpha(X^1) \geq \alpha(X^2)$ for intervals X^1, X^2 such that $X^1 \supset X^2$. Hence, when α is re-computed for each X^l in a sequence of nested intervals, the corresponding sequence $\{\alpha^l\}$, and thus also the sequence of pre-factors $\{K^l\}$, is monotonically decreasing. This explains the behavior reported in [34, Figures 1, 2] for αBB relaxations with variable α. It is not possible, however, to argue that in general $\lim_{l\to\infty} \alpha^l = 0$, which would imply a super-quadratic order of convergence, see [34, Figure 3] for a counter-example.

Lastly, note that αBB relaxations coincide with f on X when $\alpha(X) = 0$ so that the lower bound is exact in this case. In this case, convexity of f on X has been detected.

3.5 Comparison of bounding methods

In this chapter, different methods have been studied to bound the range of $\mathcal{L}$-factorable function. When the natural function is twice differentiable functions, all listed bounding methods are applicable. When this strong regularity assumption is dropped, some—such as αBB relaxations—are not defined any longer, while properties such as the convergence order of others weaken.

Table 3.2 compares the different methods by the complexity of one evaluation and their convergence order. Note that the cost of αBB relaxations is the cost to evaluate α when using the method based on Greschgorin's theorem [2], which is the most expensive step, but it must be performed only once. Once α has been determined, any further relaxation evaluation has complexity $O(n_f + n)$. Note that neither the complexity of an evaluation of a standard McCormick relaxation nor of a αBB relaxation includes the evaluation of (sub)gradients. In case of either relaxation technique, deriving a bound on the range still necessitates solving a convex optimization problem so that derivative information is necessary, too. Here, it should be pointed out that the adjoint mode of automatic differentiation [70] should be used for efficient calculation of the derivative information. A similar method is also available to efficiently calculate a subgradient of the standard McCormick relaxation [21]. Lastly note that K serves as a placeholder in the convergence

order column. For the precise forms of the convergence order pre-factor, see the results in the previous sections.

At this point, one might question the benefit of nonconstant relaxations over centered forms, which only provide a constant bound, as each method is second-order convergent. When using interval methods to approximate conservatively the feasible set of (1.1) on some $Y \in \mathbb{IC}$, this problem results

$$\begin{aligned} f^{\mathrm{L,int}}(Y) = \inf_{\mathbf{y}\in Y} \quad & \underline{f}(Y) \\ \text{s.t.} \quad & \underline{\mathbf{g}}(Y) \leq \mathbf{0}, \\ & \underline{\mathbf{h}}(Y) \leq \mathbf{0} \leq \overline{\mathbf{h}}(Y). \end{aligned} \tag{3.3}$$

Note that no optimization problem needs to be solved in order to find the lower bound

$$f^{\mathrm{L,int}}(Y) = \begin{cases} \underline{f}(Y) & \text{if } \underline{\mathbf{g}}(Y) \leq \mathbf{0}, \underline{\mathbf{h}}(Y) \leq \mathbf{0} \leq \overline{\mathbf{h}}(Y) \\ +\infty & \text{otherwise.} \end{cases}$$

Thus, (3.3) uses the constraints only to provide a certificate of infeasibility. On the other hand, constructing a relaxation of (1.1) using nonconstant convex relaxations results in

$$\begin{aligned} f^{\mathrm{L,rlx}}(Y) = \inf_{\mathbf{y}\in Y} \quad & \check{f}(\mathbf{y}) \\ \text{s.t.} \quad & \check{\mathbf{g}}(\mathbf{y}) \leq \mathbf{0}, \\ & \check{\mathbf{h}}(\mathbf{y}) \leq \mathbf{0} \leq \hat{\mathbf{h}}(\mathbf{y}), \end{aligned} \tag{3.4}$$

where each relaxation is constructed on Y. In (3.4), the relaxations of the constraints (possibly) restrict the set of permissible $\mathbf{y} \in Y$ so that $f^{\mathrm{L,rlx}}(Y) \geq \inf_{\mathbf{y}\in Y} \check{f}(\mathbf{y})$. Consequently, the wrapping effect supplies a plausible explanation why relaxations have been more successful in global optimization than second-order convergent interval bounds.

Chapter 4

Reverse propagation of McCormick relaxations

Schichl and Neumaier [153] demonstrated that factorable functions can be represented alternatively as a DAG[1], see also Section 3.1.2, and discussed how this representation can be used for calculations in interval analysis. Vu et al. [174] detailed how to propagate interval information on DAGs to improve interval bounds. Their method can utilize the information from equality and inequality constraints. We will refer to this idea as *reverse interval propagation*. In this chapter, the idea is extended to convex and concave relaxations.

The class of factorable functions encompasses most functions that can be implemented as computer programs without conditional statements. It is well-known that relaxations of factorable functions can be computed using McCormick's composition rule [118, 156]; the obtained relaxations are often referred to as *McCormick relaxations*, see Section 3.3.2. Here, it is proposed to use the DAG representation of the constraints to also propagate McCormick relaxations backward. For the benefit of the reader we provide an interpretation of relaxations in the context of constraint propagation. Suppose we partition the variables into $\mathbf{p}$ and $\mathbf{x}$. Whereas interval propagation yields a constant bound that all feasible $(\mathbf{x}, \mathbf{p})$ must satisfy, reverse McCormick propagation yields bounds that are functions of $\mathbf{p}$. For a given $\mathbf{p}$ in the domain, all $\mathbf{x}$ so that $(\mathbf{p}, \mathbf{x})$ is feasible are bounded. Figure 4.1 illustrates this interpretation. It shows that a domain (dash-dotted box) can be shrunk by interval constraint propagation to find an outer approximation of the feasible region (dotted box). However, the relaxations (solid and dashed lines) provide a tighter approximation that is a function of $\mathbf{p}$. For example, consider $\mathbf{p}_1$, for which a thick solid line shows all feasible $\mathbf{x}$. Given $\mathbf{p}_1$, the relaxations restrict $\mathbf{x}$ to the blue interval whereas the interval bounds only constrain them to the larger green interval. Furthermore, since the bounds are convex and concave functions of $\mathbf{p}$, it is tractable, for example, to calculate cheaply a reduced interval domain using affine relaxations based on the subgradients of the relaxations [121] or by minimizing and maximizing the relaxations of each x_i on the $\mathbf{p}$ domain.

The remainder of this chapter is organized as follows. Section 4.1 recapitulates the important results for reverse interval propagation from [174], which are extended to McCormick objects in Section 4.2. Section 4.3 discusses how the theoretical results from the previous section can be applied to construct and improve relaxations of implicit mappings. Section 4.4 describes how the method can be implemented and some case studies are given

[1] This representation of a factorable function is also used in the reverse mode of automatic differentiation [70].

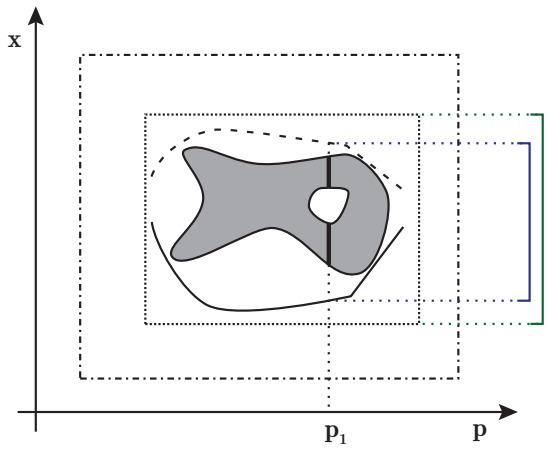


Figure 4.1: Illustration of domain reduction by reverse interval and McCormick propagation. The gray area is the set of all feasible solutions, the dash-dotted line is the original domain, the dotted line is the reduced domain using reverse interval propagation. The solid and dashed lines are relaxations of the feasible region that are functions of by $\mathbf{p}$.

in Section 4.5. Section 4.6 summarizes the results and concludes the chapter.

4.1 Reverse interval propagation

In this section, we will focus on propagating interval bounds backwards through the computational sequence, which is a particular form of a DAG. Since the reverse McCormick propagation is similar in spirit, it is very instructive to first revisit the interval case. The results, which are stated below, have been adapted from [174], though the notation is introduced here.

Definition 4.1. Consider $\mathbf{f} : D \subset \mathbb{R}^n \to \mathbb{R}^m$. Let $F^{\text{rev}} : \mathbb{I}_\emptyset D \times \mathbb{IR}^m \to \mathbb{IR}^n_\emptyset$. If for all $X \in \mathbb{I}_\emptyset D$ and $R \in \mathbb{IR}^m$ it holds that

$$\{\mathbf{x} \in X : \mathbf{f}(\mathbf{x}) \in R\} \subset \{\mathbf{x} \in F^{\text{rev}}(X, R)\} \subset X, \tag{4.1}$$

then F^{rev} is called a *reverse interval update* of $\mathbf{f}$.

Definition 4.2. Let $(\mathcal{S}, \pi_o)$ be a $\mathcal{L}$-computational sequence with n_i inputs and n_o outputs with natural interval extension $(F_\mathcal{S}, \mathfrak{D}_\mathcal{S}, \mathbb{R}^{n_o})$. Let $X \in \mathfrak{D}_\mathcal{S}$. Suppose that $V_1(X), \ldots, V_{n_f}(X)$ have been calculated according to Definition 3.12. Let $o_k^{\text{rev}} : \mathbb{I}_\emptyset B_k \times \mathbb{IR} \to \mathbb{I}_\emptyset B_k$ be a reverse interval update of o_k for each $k = n_i + 1, \ldots, n_f$. Suppose $\tilde{V}_1, \ldots, \tilde{V}_{n_f} \in \mathbb{IR}_\emptyset$ are calculated

for any $X \in \mathfrak{D}_{\mathcal{S}}$ and $R \in \mathbb{IR}^{n_o}$ by the following procedure:

$$
\begin{aligned}
&(\tilde{V}_1, \ldots, \tilde{V}_{n_f}) := (V_1(X), \ldots, V_{n_f}(X)) \\
&\pi_o(\tilde{V}_1, \ldots, \tilde{V}_{n_f}) := \pi_o(\tilde{V}_1, \ldots, \tilde{V}_{n_f}) \cap R \\
&\texttt{for } l := 1, \ldots, n_f - n_i \texttt{ do} \\
&\qquad \pi_{n_f-l+1}(\tilde{V}_1, \ldots, \tilde{V}_{n_f-l}) := o^{\mathrm{rev}}_{n_f-l+1}(\pi_{n_f-l+1}(\tilde{V}_1, \ldots, \tilde{V}_{n_f-l}), \tilde{V}_{n_f-l+1}) \\
&\texttt{end}
\end{aligned}
$$

The *reverse interval propagation* of $(\mathcal{S}, \pi_o)$ is the function $(F^{\mathrm{rev}}_{\mathcal{S}}, \mathfrak{D}_{\mathcal{S}} \times \mathbb{IR}^{n_o}, \mathbb{I}D_{\mathcal{S}})$ defined by $F^{\mathrm{rev}}_{\mathcal{S}}(X, R) \equiv (\tilde{V}_1, \ldots, \tilde{V}_{n_i})$.

Theorem 4.1. *The reverse interval propagation of $(\mathcal{S}, \pi_o)$ as given by Definition 4.2 is a reverse interval update of $(\mathbf{f}_{\mathcal{S}}, D_{\mathcal{S}}, \mathbb{R}^{n_o})$. If the reverse update of $\mathbf{o}_k$ is inclusion monotonic for each $k = n_i + 1, \ldots, n_f$ then the reverse interval propagation of $(\mathcal{S}, \pi_o)$ is inclusion monotonic.*

Proof. Finite induction yields immediately that the second inclusion in (4.1) holds.

Let $R \in \mathbb{IR}^{n_o}$ and $X \in \mathfrak{D}_{\mathcal{S}}$. If there does not exist a $\mathbf{x} \in X$ such that $\mathbf{f}_{\mathcal{S}}(\mathbf{x}) \in R$, then the first inclusion in (4.1) holds trivially.

Let $\mathbf{x} \in X$ such that $\mathbf{f}_{\mathcal{S}}(\mathbf{x}) \in R$. Then, there exists a sequence of factor values $\{v_k(\mathbf{x})\}_{k=1}^{n_f}$ with $v_1(\mathbf{x}) = x_1, \ldots, v_{n_i}(\mathbf{x}) = x_{n_i}$ and $\pi_o(v_1(\mathbf{x}), \ldots, v_{n_f}(\mathbf{x})) \in R$. Also, since $V_1, \ldots, V_{n_f}$ are inclusion functions (see Theorem 3.3), $(v_1(\mathbf{x}), \ldots, v_{n_f}(\mathbf{x})) \in (V_1(X), \ldots, V_{n_f}(X))$ so that $(v_1(\mathbf{x}), \ldots, v_{n_f}(\mathbf{x})) \in (\tilde{V}_1, \ldots, \tilde{V}_{n_f})$ prior to entering the loop. In the following, let $\tilde{V}_k^l$ denote the value of $\tilde{V}_k$ for the given X and R after the lth reverse update, $l = 1, \ldots, n_f - n_i$. Since $o^{\mathrm{rev}}_{n_f}$ is a reverse interval update, it follows that $(v_1(\mathbf{x}), \ldots, v_{n_f-1}(\mathbf{x})) \in (\tilde{V}_1^1, \ldots, \tilde{V}_{n_f-1}^1)$. Finite induction yields that $(v_1(\mathbf{x}), \ldots, v_{n_i}(\mathbf{x})) \in (\tilde{V}_1^{n_f-n_i}, \ldots, \tilde{V}_{n_i}^{n_f-n_i}) \equiv F^{\mathrm{rev}}_{\mathcal{S}}(X, R)$. Thus, $\mathbf{x} \in F^{\mathrm{rev}}_{\mathcal{S}}(X, R)$ and the first inclusion in (4.1) holds.

Assume now that o^{rev}_k is inclusion monotonic for each $k = n_i + 1, \ldots, n_f$. Let $X^1, X^2 \in \mathfrak{D}_{\mathcal{S}}$ with $X^1 \subset X^2$ and $R^1, R^2 \in \mathbb{IR}^{n_o}$ with $R^1 \subset R^2$. Then, $(\tilde{V}_1(X^1, R^1), \ldots, \tilde{V}_{n_f}(X^1, R^1)) \subset (\tilde{V}_1(X^2, R^2), \ldots, \tilde{V}_{n_f}(X^2, R^2))$ prior to entering the loop. Since $\mathbf{o}^{\mathrm{rev}}_{n_f}$ is inclusion monotonic, $(\tilde{V}_1^1(X^1, R^1), \ldots, \tilde{V}_{n_f}^1(X^1, R^1)) \subset (\tilde{V}_1^1(X^2, R^2), \ldots, \tilde{V}_{n_f}^1(X^2, R^2))$. Using finite induction over l yields that $(\tilde{V}_1^{n_f-n_i}(X^1, R^1), \ldots, \tilde{V}_{n_i}^{n_f-n_i}(X^1, R^1)) \subset (\tilde{V}_1^{n_f-n_i}(X^2, R^2), \ldots, \tilde{V}_{n_i}^{n_f-n_i}(X^2, R^2))$. Thus, it follows that $F^{\mathrm{rev}}_{\mathcal{S}}(X^1, R^1) \subset F^{\mathrm{rev}}_{\mathcal{S}}(X^2, R^2)$. □

Next, we will present a result very closely related to Theorem 4.1 that relies more on standard concepts from interval analysis.

Theorem 4.2. *Consider $(\mathcal{S}, \pi_o)$ and assume that for each $k = n_i + 1, \ldots, n_f$, the reverse interval update of o_k is inclusion monotonic. Define $\mathbf{f}^{\mathrm{rev}}_{\mathcal{S}} : D \times \mathbb{R}^{n_o} \to \mathbb{R}^{n_i}_{\emptyset}$ for each $\mathbf{x} \in D$ and $\mathbf{r} \in \mathbb{R}^{n_o}$ by*

$$
\mathbf{f}^{\mathrm{rev}}_{\mathcal{S}}(\mathbf{x}, \mathbf{r}) = \begin{cases} \mathbf{x} & \text{if } \mathbf{f}_{\mathcal{S}}(\mathbf{x}) = \mathbf{r}, \\ \mathrm{NaN} & \text{otherwise.} \end{cases} \tag{4.2}
$$

Then, $F^{\mathrm{rev}}_{\mathcal{S}}$ is an inclusion function of $\mathbf{f}^{\mathrm{rev}}_{\mathcal{S}}$ on $\mathfrak{D}_{\mathcal{S}} \times \mathbb{IR}^{n_o}$.

Proof. Let $\mathbf{r} \in \bar{\mathbb{R}}^{n_o}$. First, consider $\mathbf{x} \in D$ so that $\mathbf{f}_{\mathcal{S}}(\mathbf{x}) = \mathbf{r}$. Since $F_{\mathcal{S}}$ is an interval extension of $\mathbf{f}_{\mathcal{S}}$, each factor is a degenerate interval after the forward evaluation with $V_k([\mathbf{x},\mathbf{x}]) = [v_k(\mathbf{x}), v_k(\mathbf{x})]$. Since $\mathbf{f}_{\mathcal{S}}(\mathbf{x}) = \mathbf{r}$, the intersections during the reverse interval propagation return the degenerate intervals so that it is clear that $F_{\mathcal{S}}^{\text{rev}}$ is an interval extension of $\mathbf{f}_{\mathcal{S}}^{\text{rev}}$ for such $[\mathbf{x},\mathbf{x}]$. If $\mathbf{x} \in D$ such that $\mathbf{f}_{\mathcal{S}}(\mathbf{x}) \neq \mathbf{r}$ then $\pi_o(\tilde{V}_1, \ldots, \tilde{V}_{n_f}) := \pi_o(\tilde{V}_1, \ldots, \tilde{V}_{n_f}) \cap R$ results in $(\tilde{V}_1 \times \cdots \times \tilde{V}_{n_f}) = \emptyset$. Any interval calculation involving empty intervals again yields empty intervals so that $F_{\mathcal{S}}^{\text{rev}}([\mathbf{x},\mathbf{x}],[\mathbf{r},\mathbf{r}]) = \emptyset = [\text{NaN},\text{NaN}] = [\mathbf{f}_{\mathcal{S}}^{\text{rev}}(\mathbf{x},\mathbf{r}), \mathbf{f}_{\mathcal{S}}^{\text{rev}}(\mathbf{x},\mathbf{r})]$. Thus, $F_{\mathcal{S}}^{\text{rev}}$ is an interval extension of $\mathbf{f}_{\mathcal{S}}^{\text{rev}}$. Inclusion monotonicity of $F_{\mathcal{S}}^{\text{rev}}$ has been established in Theorem 4.1. The assertion follows then from Theorem 3.1. □

4.1.1 Reverse interval updates of binary operations

It is sufficient to study addition and multiplication. Subtraction and division can be considered by using the negative or reciprocal operators, which are in $\mathcal{L}$, the library of univariate functions.

Lemma 4.1. *Consider* $(+, \mathbb{R}^2, \mathbb{R})$. *The function* $(+^{\text{rev}}, \mathbb{IR}^2_{\emptyset} \times \mathbb{I}\bar{\mathbb{R}}, \mathbb{IR}^2_{\emptyset})$ *defined for all* $X, Y \in \mathbb{IR}_{\emptyset}$ *and* $R \in \mathbb{I}\bar{\mathbb{R}}$ *by*

$$+^{\text{rev}}((X,Y),R) = (R - Y, R - X) \cap (X,Y)$$

is an inclusion monotonic reverse interval update of $(+, \mathbb{R}^2, \mathbb{R})$.

Proof. Let $X, Y \in \mathbb{IR}_{\emptyset}$, $R \in \mathbb{I}\bar{\mathbb{R}}$. If $+(X,Y) \cap R = \emptyset$, then $\nexists (x,y,r) \in X \times Y \times R : x + y = r$. Thus, $r - y \notin X$ for all $(y,r) \in Y \times R$ so that $(R - Y) \cap X = \emptyset$. Similarly, $(R - X) \cap Y = \emptyset$ so that (4.1) holds trivially.

Otherwise, pick $(x,y) \in X \times Y$ so that $x + y \in R$. Since $\underline{r} \leq x + y \leq \overline{r}$ and $(x,y) \in ([\underline{x},\overline{x}],[\underline{y},\overline{y}])$, it follows that $x \geq \underline{r} - y \geq \underline{r} - \overline{y}$ and $x \leq \overline{r} - y \leq \overline{r} - \underline{y}$ and that $y \geq \underline{r} - x \geq \underline{r} - \overline{x}$ and $y \leq \overline{r} - x \leq \overline{r} - \underline{x}$ so that $(x,y) \in +^{\text{rev}}((X,Y),R)$. Thus, (4.1) holds.

Inclusion monotonicity follows directly from inclusion monotonicity of subtraction and intersection. □

While it may appear to be more advantageous to use

$$+^{\text{rev}}((X,Y),R) = (R - Y, R - ((R - Y) \cap X)) \cap (X,Y)$$

in fact there is no benefit, as the following argument shows. Let $(X',Y') = +^{\text{rev}}((X,Y),R)$ and note that

$$\begin{aligned}
\underline{x}' &= \max(\underline{r} - \overline{y}, \underline{x}) \\
\overline{x}' &= \min(\overline{r} - \underline{y}, \overline{x}) \\
\underline{y}' &= \max(\underline{r} - \min(\overline{r} - \underline{y}, \overline{x}), \underline{y}) = \max(\underline{r} - \overline{r} + \underline{y}, \underline{r} - \overline{x}, \underline{y}) = \max(\underline{r} - \overline{x}, \underline{y}) \\
\overline{y}' &= \min(\overline{r} - \max(\underline{r} - \overline{y}, \underline{x}), \overline{y}) = \min(\overline{r} - \underline{r} + \overline{y}, \overline{r} - \underline{x}, \overline{y}) = \min(\overline{r} - \underline{x}, \overline{y}),
\end{aligned}$$

where we have used the fact that $\underline{r} \leq \overline{r}$.

Next, we will study the reverse interval update for multiplication. Note that $(\underline{x}, \overline{x}) \equiv \{x \in \mathbb{R} : \underline{x} < x < \overline{x}\}$ denotes an *open interval* and that for two sets A, B, the *relative complement* is denoted by $A \setminus B \equiv \{x \in A : x \notin B\}$.

Proposition 4.1. *[127, Proposition 4.2.1] Define the* Gauss-Seidel operator $(\Gamma, \mathbb{IR}_\emptyset \times \mathbb{I}\bar{\mathbb{R}} \times \mathbb{IR}_\emptyset, \mathbb{IR}_\emptyset)$ *for all* $X, Y \in \mathbb{IR}_\emptyset$ *and* $R \in \mathbb{I}\bar{\mathbb{R}}$ *by*

$$\Gamma(X, R, Y) = \text{hull}\{y \in Y : \exists x \in X, r \in R : xy = r\}.$$

Then,

$$\Gamma(X, R, Y) = \begin{cases} \left(R \times \frac{1}{X}\right) \cap Y & \text{if } 0 \notin X, \\ \text{hull}(Y \setminus (\underline{r}/\underline{x}, \underline{r}/\overline{x})) & \text{if } \underline{r} > 0,\ 0 \in X, \\ \text{hull}(Y \setminus (\overline{r}/\overline{x}, \overline{r}/\underline{x})) & \text{if } \underline{r} < 0,\ 0 \in X, \\ Y & \text{otherwise.} \end{cases}$$

Lemma 4.2. *Consider* $(\times, \mathbb{R}^2, \mathbb{R})$. *The function* $(\times^{\text{rev}}, \mathbb{IR}^2_\emptyset \times \mathbb{I}\bar{\mathbb{R}}, \mathbb{IR}^2_\emptyset)$ *defined for all* $X, Y \in \mathbb{IR}_\emptyset$ *and* $R \in \mathbb{I}\bar{\mathbb{R}}$ *by*

$$\times^{\text{rev}}((X, Y), R) = (\Gamma(Y, R, X), \Gamma(\Gamma(Y, R, X), R, Y))$$

is an inclusion monotonic reverse interval update of $(\times, \mathbb{R}^2, \mathbb{R})$.

Proof. Let $X, Y \in \mathbb{IR}_\emptyset$, $R \in \mathbb{I}\bar{\mathbb{R}}$. If $\times(X, Y) \cap R = \emptyset$, there does not exist a $x \in X$, $y \in Y$ so that $xy \in R$, i.e., $\Gamma(Y, R, X) = \emptyset$. Thus, $\Gamma(\Gamma(Y, R, X), R, Y) = \emptyset$ so that $\times^{\text{rev}}((X, Y), R) = \emptyset$ and (4.1) holds trivially.

Otherwise, pick $(x, y) \in X \times Y$ so that $xy \in R$. Note that $\Gamma(Y, R, X) = \{\tilde{x} \in X : \exists \tilde{y} \in Y, z \in R : \tilde{x}\tilde{y} = z\}$, and hence $x \in \Gamma(Y, R, X) \subset X$. Likewise, $\Gamma(\Gamma(Y, R, X), R, Y) = \{\tilde{y} \in Y : \exists \tilde{x} \in \Gamma(Y, R, X), z \in R : \tilde{x}\tilde{y} = z\}$, hence $y \in \Gamma(\Gamma(Y, R, X), R, Y)$. Thus, $(x, y) \in \times^{\text{rev}}((X, Y), R)$ so that the first inclusion (4.1) holds. The second inclusion follows from [127, 4.3.2].

Inclusion monotonicity follows directly from [127, 4.3.2]. □

Note that $\times^{\text{rev}}((X, Y), R) = (\Gamma(\Gamma(X, R, Y), R, X), \Gamma(X, R, Y))$ is an alternative reverse interval update of $(\times, \mathbb{R}^2, \mathbb{R})$. In particular, the sequential update of X and then Y provides a benefit here whereas for $(+, \mathbb{R}^2, \mathbb{R})$ it does not yield additional information.

4.1.2 Reverse interval updates of univariate functions

Lemma 4.3. *Let* $B \subset \mathbb{R}$ *and consider an injective continuous function* $(u, B, \mathbb{R}) \in \mathcal{L}$. *Suppose that* $(u, \mathbb{I}B, \mathbb{IR})$ *is exact, i.e., it maps to the image of the real-valued function for any* $X \in \mathbb{I}B$. *The function* $(u^{\text{rev}}, \mathbb{I}_\emptyset B \times \mathbb{I}\bar{\mathbb{R}}, \mathbb{IR}_\emptyset)$ *defined for all* $X \in \mathbb{I}_\emptyset B$ *and* $R \in \mathbb{I}\bar{\mathbb{R}}$ *by*

$$u^{\text{rev}}(X, R) = [\min(u^{-1}(\underline{t}), u^{-1}(\overline{t})), \max(u^{-1}(\underline{t}), u^{-1}(\overline{t}))],$$

where $T = R \cap u(X)$, *is an inclusion monotonic reverse interval update of* $(u, B, \mathbb{R})$.

Proof. Let $X \in \mathbb{I}_\emptyset B$. Suppose that $T = \emptyset$, in which case $u^{\text{rev}}(X, R) = \emptyset$. Then, since $(u, \mathbb{I}B, \mathbb{IR})$ is an inclusion function, there does not exist an $x \in X$ so that $u(x) \in R$.

Otherwise, since $(u, B, \mathbb{R})$ is continuous and injective, it is invertible on $u(B)$ and u^{-1} is continuous [145, Thm. 4.17]. Since $T \subset u(B)$, $u^{-1}(\underline{t})$ and $u^{-1}(\bar{t})$ are defined. Since u is invertible on X it is bijective as a mapping into T. This implies that u^{-1} is also injective on T. Note that $T \in \mathbb{IR}$. u^{-1} is monotonic on T so that $\underline{t} < t < \bar{t}$ implies that either $u^{-1}(\underline{t}) < u^{-1}(t) < u^{-1}(t))$ or $u^{-1}(\underline{t}) > u^{-1}(t) > u^{-1}(\bar{t})$. Thus, $x \in X$ so that $u(x) \in T$ implies that $x \in [\min(u^{-1}(\underline{t}), u^{-1}(\bar{t})), \max(u^{-1}(\underline{t}), u^{-1}(\bar{t}))] \subset X$ where the inclusion follows from $T \subset u(X)$.

Inclusion monotonicity follows directly from the monotonicity of $(u^{-1}, u(B), B)$. □

Remark 4.1. Lemma 4.3 can be used to define the reverse interval update of $-(\cdot)$, $(\cdot)^n$ for odd $n \in \mathbb{N}$, exp, log, $\sqrt{\cdot}$, etc. It is also applicable to $\frac{1}{(\cdot)}$ if B is restricted to either the negative or positive reals.

Lemma 4.4. *Let $n \in \mathbb{N}$ be even. Consider $(u, \mathbb{R}, \mathbb{R})$ where $u(x) = x^n$. The function $(u^{\mathrm{rev}}, \mathbb{IR}_\emptyset \times \mathbb{IR}, \mathbb{IR}_\emptyset)$ defined for all $X \in \mathbb{IR}_\emptyset$ and $R \in \mathbb{IR}$ by*

$$u^{\mathrm{rev}}(X, R) = \operatorname{hull}\left(X \cap \left[-\sqrt[n]{\bar{t}}, -\sqrt[n]{\underline{t}}\right], X \cap \left[\sqrt[n]{\underline{t}}, \sqrt[n]{\bar{t}}\right]\right)$$

where $T = R \cap u(X)$ is an inclusion monotonic reverse interval update of $(u, \mathbb{R}, \mathbb{R})$.

Proof. Let $X \in \mathbb{IR}_\emptyset$. Suppose that $T = \emptyset$, in which case $u^{\mathrm{rev}}(X, R) = \emptyset$. Then, since $(u, \mathbb{IR}, \mathbb{IR})$ is an inclusion function, there does not exist an $x \in X$ so that $u(x) \in R$.

Otherwise, since the equation $u(x) = r$ has two solutions for any positive r, namely $x = \sqrt[n]{r}$ and $x = -\sqrt[n]{r}$, any

$$\tilde{x} \in \tilde{X} = \{x \in X : \exists r \in R, x = \sqrt[n]{r} \vee x = -\sqrt[n]{r}\}$$

will satisfy $u(\tilde{x}) \in T$. The argument still holds for $r = 0$. Intersecting $\left[-\sqrt[n]{\bar{t}}, -\sqrt[n]{\underline{t}}\right]$ or $\left[\sqrt[n]{\underline{t}}, \sqrt[n]{\bar{t}}\right]$ with X will not discard any $\tilde{x} \in \tilde{X}$. Since these intervals may be disjoint, constructing the interval hull will yield an interval.

Inclusion monotonicity follows directly since $(\sqrt[n]{\cdot}, [0, +\infty), \mathbb{R})$ is monotonic and from the inclusion monotonicity of the intersection and hull operators. □

Note that with a construction similar to Lemma 4.4 it is possible to find a reverse interval update of the absolute value function.

4.2 Reverse McCormick propagation

In this section, the ideas for reverse interval propagation are extended to McCormick objects. Again, the enclosure property will be established, but also coherent concavity and inclusion monotonicity of the resulting relaxations will be proved.

Definition 4.3. Suppose $\mathbf{f} : D \subset \mathbb{R}^n \to \mathbb{R}^m$. Consider $\mathcal{F}^{\text{rev}} : \mathbb{M}_\emptyset D \times \bar{\mathbb{M}}\mathbb{R}^m \to \mathbb{M}\mathbb{R}^n_\emptyset$. If for all $\mathcal{X} \in \mathbb{M}_\emptyset D$ and $\mathcal{R} \in \bar{\mathbb{M}}\mathbb{R}^m$ it holds that

$$\{\mathbf{x} \in \text{Enc}(\mathcal{X}) : \mathbf{f}(\mathbf{x}) \in \text{Enc}(\mathcal{R})\} \subset \{\mathbf{x} \in \text{Enc}(\mathcal{F}^{\text{rev}}(\mathcal{X}, \mathcal{R}))\} \tag{4.3}$$

and $\mathcal{F}^{\text{rev}}(\mathcal{X}, \mathcal{R}) \subset \mathcal{X}$, then $\mathcal{F}^{\text{rev}}$ is called a *reverse McCormick update* of $\mathbf{f}$.

Definition 4.4. Let $(\mathcal{S}, \pi_o)$ be a $\mathcal{L}$-computational sequence with n_i inputs and n_o outputs with natural McCormick extension $(\mathcal{F}_\mathcal{S}, \mathcal{D}_\mathcal{S}, \mathbb{R}^{n_o})$. Let $\mathcal{X} \in \mathcal{D}_\mathcal{S}$. Suppose $\mathcal{V}_1(\mathcal{X}), \ldots, \mathcal{V}_{n_f}(\mathcal{X})$ have been calculated according to Definition 3.22. Let $o_k^{\text{rev}} : \mathbb{M}_\emptyset B_k \times \bar{\mathbb{M}}\mathbb{R} \to \mathbb{M}_\emptyset B_k$ be a reverse McCormick update of o_k for each $k = n_i + 1, \ldots, n_f$. Suppose $\tilde{\mathcal{V}}_1, \ldots, \tilde{\mathcal{V}}_{n_f} \in \mathbb{M}\mathbb{R}_\emptyset$ are calculated for any $\mathcal{X} \in \mathcal{D}_\mathcal{S}$ and $\mathcal{R} \in \bar{\mathbb{M}}\mathbb{R}^{n_o}$ by the following procedure:

$$\begin{aligned}
&(\tilde{\mathcal{V}}_1, \ldots, \tilde{\mathcal{V}}_{n_f}) := (\mathcal{V}_1(\mathcal{X}), \ldots, \mathcal{V}_{n_f}(\mathcal{X})) \\
&\pi_o(\tilde{\mathcal{V}}_1, \ldots, \tilde{\mathcal{V}}_{n_f}) := \pi_o(\tilde{\mathcal{V}}_1, \ldots, \tilde{\mathcal{V}}_{n_f}) \cap \mathcal{R} \\
&\texttt{for } l := 1, \ldots, n_f - n_i \texttt{ do} \\
&\qquad \pi_{n_f - l + 1}(\tilde{\mathcal{V}}_1, \ldots, \tilde{\mathcal{V}}_{n_f - l}) := o^{\text{rev}}_{n_f - l + 1}(\pi_{n_f - l + 1}(\tilde{\mathcal{V}}_1, \ldots, \tilde{\mathcal{V}}_{n_f - l}), \tilde{\mathcal{V}}_{n_f - l + 1}) \\
&\texttt{end}
\end{aligned}$$

The *reverse McCormick propagation* of $(\mathcal{S}, \pi_o)$ is the function $(\mathcal{F}^{\text{rev}}_\mathcal{S}, \mathcal{D}_\mathcal{S} \times \bar{\mathbb{M}}\mathbb{R}^{n_o}, \mathbb{M}_\emptyset D_\mathcal{S})$ defined for any $\mathcal{X} \in \mathcal{D}_\mathcal{S}$ and $\mathcal{R} \in \bar{\mathbb{M}}\mathbb{R}^{n_o}$ by $\mathcal{F}^{\text{rev}}_\mathcal{S}(\mathcal{X}, \mathcal{R}) \equiv (\tilde{\mathcal{V}}_1, \ldots, \tilde{\mathcal{V}}_{n_i})$.

Theorem 4.3. *The reverse McCormick propagation of $(\mathcal{S}, \pi_o)$ as given by Definition 4.4 is a reverse McCormick update of $(\mathbf{f}_\mathcal{S}, D_\mathcal{S}, \mathbb{R}^{n_o})$.*

Proof. Let $\mathcal{R} \in \bar{\mathbb{M}}\mathbb{R}^m$ and $\mathcal{X} \in \mathcal{D}_\mathcal{S}$. Finite induction yields immediately that $\mathcal{F}^{\text{rev}}(\mathcal{X}, \mathcal{R}) \subset \mathcal{X}$. If there does not exist $\mathbf{x} \in \text{Enc}(\mathcal{X})$ such that $\mathbf{f}_\mathcal{S}(\mathbf{x}) \in \text{Enc}(\mathcal{R})$, then (4.3) holds trivially.

Let $\mathbf{x} \in \mathcal{X}$ satisfy $\mathbf{f}_\mathcal{S}(\mathbf{x}) \in \text{Enc}(\mathcal{R})$. Then, there exists a sequence of factor values $\{v_k(\mathbf{x})\}_{k=1}^{n_f}$ with $v_1(\mathbf{x}) = x_1, \ldots, v_{n_i}(\mathbf{x}) = x_{n_i}$ and $\pi_o(v_1(\mathbf{x}), \ldots, v_{n_f}(\mathbf{x})) \in \text{Enc}(\mathcal{R})$. Also, since $\mathcal{V}_1, \ldots, \mathcal{V}_{n_f}$ are relaxation functions, $(v_1(\mathbf{x}), \ldots, v_{n_f}(\mathbf{x})) \in \text{Enc}((\mathcal{V}_1(\mathcal{X}), \ldots, \mathcal{V}_{n_f}(\mathcal{X})))$ so that $(v_1(\mathbf{x}), \ldots, v_{n_f}(\mathbf{x})) \in \text{Enc}((\tilde{\mathcal{V}}_1, \ldots, \tilde{\mathcal{V}}_{n_f}))$ prior to entering the loop.

In the following, let $\tilde{\mathcal{V}}_k^l$ denote the value of $\tilde{\mathcal{V}}_k$ for the given $\mathcal{X}$ and $\mathcal{R}$ after the lth reverse update, $l = 1, \ldots, n_f - n_i$. Since $o^{\text{rev}}_{n_f}$ is a reverse McCormick update, it follows that $(v_1(\mathbf{x}), \ldots, v_{n_f - 1}(\mathbf{x})) \in \text{Enc}(\tilde{\mathcal{V}}_1^1, \ldots, \tilde{\mathcal{V}}_{n_f - 1}^1)$. Finite induction yields that $(v_1(\mathbf{x}), \ldots, v_{n_i}(\mathbf{x})) \in \text{Enc}((\tilde{\mathcal{V}}_1^{n_f - n_i}, \ldots, \tilde{\mathcal{V}}_{n_i}^{n_f - n_i})) \equiv \text{Enc}(\mathcal{F}^{\text{rev}}_\mathcal{S}(\mathcal{X}, \mathcal{R}))$. It thus follows that $\mathbf{x} \in \text{Enc}(\mathcal{F}^{\text{rev}}_\mathcal{S}(\mathcal{X}, \mathcal{R}))$ and (4.3) holds. □

Lemma 4.5. *Consider $(\mathcal{S}, \pi_o)$ and assume that for each $k = n_i + 1, \ldots, n_f$, the reverse McCormick update of o_k is coherently concave and inclusion monotonic on $\mathbb{M}_\emptyset B_k \times \bar{\mathbb{M}}\mathbb{R}$. Then, $\mathcal{F}^{\text{rev}}_\mathcal{S}$ is coherently concave and inclusion monotonic on $\mathcal{D}_\mathcal{S} \times \bar{\mathbb{M}}\mathbb{R}^{n_o}$.*

Proof. Compositions of coherently concave and inclusion monotonic functions are coherently concave and inclusion monotonic [155, Lemma 2.4.15]. The result thus follows from finite induction, analogous to the proof of Theorem 4.3. □

Theorem 4.4. *Consider $(\mathcal{S}, \pi_o)$ and assume that for each $k = n_i + 1, \ldots, n_f$, the reverse McCormick update of o_k is coherently concave and inclusion monotonic. Then, $\mathcal{F}_{\mathcal{S}}^{\text{rev}}$ is a relaxation function of $\mathbf{f}_{\mathcal{S}}^{\text{rev}}$ on $\mathcal{D}_{\mathcal{S}} \times \mathbb{MIR}^{n_o}$.*

Proof. Let $\mathbf{r} \in \mathbb{IR}^{n_o}$. First, consider $\mathbf{x} \in D$ so that $\mathbf{f}_{\mathcal{S}}(\mathbf{x}) = \mathbf{r}$. It is clear that $\mathcal{F}_{\mathcal{S}}^{\text{rev}}$ is a McCormick extension of $\mathbf{f}_{\mathcal{S}}^{\text{rev}}$ for such $([\mathbf{x},\mathbf{x}],[\mathbf{x},\mathbf{x}])$ since $\mathcal{F}_{\mathcal{S}}$ is a McCormick extension of $\mathbf{f}_{\mathcal{S}}$ and $o_k^{\text{rev}}(\mathcal{B},\mathcal{R}) \subset \mathcal{B}$ for all $(\mathcal{B},\mathcal{R}) \in \mathbb{M}_{\emptyset} B_k \times \mathbb{MIR}$ by definition. If $\mathbf{x} \in D$ such that $\mathbf{f}_{\mathcal{S}}(\mathbf{x}) \neq \mathbf{r}$ then $\pi_o(\tilde{\mathcal{V}}_1, \ldots, \tilde{\mathcal{V}}_{n_f}) := \pi_o(\tilde{\mathcal{V}}_1, \ldots, \tilde{\mathcal{V}}_{n_f}) \cap ([\mathbf{r},\mathbf{r}],[\mathbf{r},\mathbf{r}])$ results in $(\tilde{\mathcal{V}}_1, \ldots, \tilde{\mathcal{V}}_{n_f}) = \emptyset$. Any calculation involving empty McCormick objects again yields empty McCormick objects so that $\mathcal{F}_{\mathcal{S}}^{\text{rev}}(([\mathbf{x},\mathbf{x}],[\mathbf{x},\mathbf{x}]),([\mathbf{r},\mathbf{r}],[\mathbf{r},\mathbf{r}])) = \emptyset$. Thus, $\mathcal{F}_{\mathcal{S}}^{\text{rev}}$ is a McCormick extension of $\mathbf{f}_{\mathcal{S}}^{\text{rev}}$. The assertion follows from Lemma 4.5 in conjunction with Theorem 3.8. □

4.2.1 Reverse McCormick updates of binary operations

Lemma 4.6. *Consider $(+, \mathbb{R}^2, \mathbb{R})$ and its relaxation function $(+, \mathbb{MIR}^2, \mathbb{MIR})$. The function $(+^{\text{rev}}, \mathbb{MIR}^2_{\emptyset} \times \mathbb{MIR}, \mathbb{MIR}^2_{\emptyset})$ defined for all $\mathcal{X}, \mathcal{Y} \in \mathbb{MIR}_{\emptyset}$ and $\mathcal{R} \in \mathbb{MIR}$ by*

$$+^{\text{rev}}((\mathcal{X},\mathcal{Y}),\mathcal{R}) = (\mathcal{R} - \mathcal{Y}, \mathcal{R} - \mathcal{X}) \cap (\mathcal{X},\mathcal{Y})$$

is a reverse McCormick update of $(+, \mathbb{R}^2, \mathbb{R})$.

Proof. Let $\mathcal{X}, \mathcal{Y} \in \mathbb{MIR}_{\emptyset}$, $\mathcal{R} \in \mathbb{MIR}$. If $\text{Enc}(+(\mathcal{X},\mathcal{Y}) \cap \mathcal{R}) = \emptyset$, then $\nexists (x,y,r) \in \text{Enc}((\mathcal{X},\mathcal{Y},\mathcal{R})) : r - y = x$. Thus, $r - y \notin \text{Enc}(\mathcal{X})$ for all $(y,r) \in \text{Enc}((\mathcal{Y},\mathcal{R}))$ so that $\text{Enc}(\mathcal{R} - \mathcal{Y}) \cap \text{Enc}(\mathcal{X}) = \emptyset$. Similarly, $\text{Enc}(\mathcal{R} - \mathcal{X}) \cap \text{Enc}(\mathcal{Y}) = \emptyset$ so that (4.3) holds trivially.

Otherwise, pick $(x,y) \in \text{Enc}(\mathcal{X}) \times \text{Enc}(\mathcal{Y})$ so that $x + y \in \text{Enc}(\mathcal{R})$. Since $\check{\phi} \leq x + y \leq \hat{\phi}$ and $(x,y) \in ([\check{x},\hat{x}],[\check{y},\hat{y}])$, it follows that $x \geq \check{\phi} - y \geq \check{\phi} - \hat{y}$ and $x \leq \hat{\phi} - \check{y} \leq \hat{\phi} - \check{y}$ and that $y \geq \check{\phi} - x \geq \check{\phi} - \hat{x}$ and $y \leq \hat{\phi} - x \leq \hat{\phi} - \check{x}$ so that $(x,y) \in \text{Enc}(+^{\text{rev}}((\mathcal{X},\check{\mathcal{Y}}),\mathcal{R}))$. Thus, (4.3)$\check{}$ holds. □

Definition 4.5. Define an *extension of the Gauss-Seidel operator* to $\mathbb{MIR}$, denoted as $\mathcal{G} : \mathbb{MIR}_{\emptyset} \times \mathbb{MIR} \times \mathbb{MIR}_{\emptyset} \to \mathbb{MIR}_{\emptyset}$, for all $\mathcal{X}, \mathcal{Y} \in \mathbb{MIR}_{\emptyset}$ and $\mathcal{R} \in \mathbb{MIR}$ by $(\mathcal{G}(\mathcal{X},\mathcal{R},\mathcal{Y}))^B = \Gamma(X^B, R^B, Y^B)$ and

$$(\mathcal{G}(\mathcal{X},\mathcal{R},\mathcal{Y}))^C = \begin{cases} (\mathcal{R}' \times \frac{1}{\mathcal{X}'})^C \cap (Y')^C & \text{if } 0 \notin X^B, \\ \Gamma(X^B, R^B, Y^B) \cap (Y')^C & \text{if } 0 \in X^B, 0 \notin R^B, \\ (Y')^C & \text{otherwise,} \end{cases}$$

where $\mathcal{X}' = (X^B, X^B \cap X^C)$, $\mathcal{Y}' = (Y^B, Y^B \cap Y^C)$ and $\mathcal{R}' = (R^B, R^B \cap R^C)$.

Lemma 4.7. *Suppose $\mathcal{X}, \mathcal{Y} \in \mathbb{MIR}_{\emptyset}$, $\mathcal{R} \in \mathbb{MIR}$. Then, $\mathcal{G}(\mathcal{X},\mathcal{R},\mathcal{Y}) \subset \mathcal{B}$ and*

$$\text{Enc}(\mathcal{G}(\mathcal{X},\mathcal{R},\mathcal{Y})) \supset \{y \in \text{Enc}(\mathcal{Y}) : \exists x \in \text{Enc}(\mathcal{X}), r \in \text{Enc}(\mathcal{R}) : xy = r\}. \tag{4.4}$$

Proof. $\Gamma(X^B, R^B, B^B) \subset B^B$ follows from [127, 4.3.2] and it is also clear that $(\mathcal{G}(\mathcal{X},\mathcal{R},\mathcal{Y}))^C \subset (Y')^C$, hence $\mathcal{G}(\mathcal{X},\mathcal{R},\mathcal{Y}) \subset \mathcal{Y}' \subset \mathcal{Y}$. Proposition 4.1 already established that

$$\Gamma(X^B, R^B, Y^B) = \text{hull}\{y \in B^B : \exists a \in X^B, r \in R^B : xy = r\}.$$

Next, note that

$$\begin{aligned}\operatorname{hull}\{y \in Y^B : \exists x \in X^B, r \in R^B : xy = r\} \supset \\ \{y \in \operatorname{Enc}(\mathcal{Y}) : \exists x \in \operatorname{Enc}(\mathcal{X}), r \in \operatorname{Enc}(\mathcal{R}) : xy = r\}\end{aligned}$$

since $\operatorname{Enc}(\mathcal{X}) \subset X^B$, $\operatorname{Enc}(\mathcal{Y}) \subset Y^B$, and $\operatorname{Enc}(\mathcal{R}) \subset R^B$. Therefore, (4.4) holds for the second and third case. Establishing $(\mathcal{G}(\mathcal{X}, \mathcal{R}, \mathcal{Y}))^C \supset \operatorname{hull}\{y \in \operatorname{Enc}(\mathcal{Y}) : \exists x \in \operatorname{Enc}(\mathcal{X}), r \in \operatorname{Enc}(\mathcal{R}) : xy = r\}$ is sufficient to show that (4.4) holds in the first case.

Suppose that $0 \notin X^B$. Consider $y \in Y^C$ such that $\exists x \in (X')^C, r \in (R')^C$ with $xy = r$, noting that $x \neq 0$ by assumption. If such y does not exist then $\{y \in \operatorname{Enc}(\mathcal{Y}) : \exists x \in \operatorname{Enc}(\mathcal{X}), r \in \operatorname{Enc}(\mathcal{R}) : xy = r\} = \emptyset$ and (4.4) holds trivially. If such y exists, then $y = r \times \frac{1}{x}$. Also, $\frac{1}{\mathcal{X}'}$ exists and $\frac{1}{x} \in \operatorname{Enc}(\frac{1}{\mathcal{X}'})$. Since $r \times \frac{1}{x} \in (\mathcal{R}' \times \frac{1}{\mathcal{X}'})^C$, $(\mathcal{G}(\mathcal{X}, \mathcal{R}, \mathcal{Y}))^C \supset \{y \in \operatorname{Enc}(\mathcal{Y}) : \exists x \in \operatorname{Enc}(\mathcal{X}), r \in \operatorname{Enc}(\mathcal{R}) : xy = r\}$. □

Lemma 4.8. *Consider* $(\times, \mathbb{R}^2, \mathbb{R})$ *and its relaxation function* $(\times, \mathbb{MR}^2, \mathbb{MR})$. *The function* $(\times^{\text{rev}}, \mathbb{MR}^2_\emptyset \times \bar{\mathbb{MR}}, \mathbb{MR}^2_\emptyset)$ *defined for all* $\mathcal{X}, \mathcal{Y} \in \mathbb{MR}_\emptyset$ *and* $\mathcal{R} \in \bar{\mathbb{MR}}$ *by*

$$\times^{\text{rev}}((\mathcal{X}, \mathcal{Y}), \mathcal{R}) = (\mathcal{G}(\mathcal{Y}, \mathcal{R}, \mathcal{X}), \mathcal{G}(\mathcal{G}(\mathcal{Y}, \mathcal{R}, \mathcal{X}), \mathcal{R}, \mathcal{Y}))$$

is a reverse McCormick update of $(\times, \mathbb{R}^2, \mathbb{R})$.

Proof. Let $\mathcal{X}, \mathcal{Y} \in \mathbb{MR}_\emptyset$, $\mathcal{R} \in \bar{\mathbb{MR}}$. If $\times(\mathcal{X}, \mathcal{Y}) \cap \mathcal{R} = \emptyset$, there does not exist $x \in \operatorname{Enc}(\mathcal{X})$, $y \in \operatorname{Enc}(\mathcal{Y})$ so that $xy \in \operatorname{Enc}(\mathcal{R})$. Thus, (4.3) holds trivially.

Otherwise, pick $(x, y) \in \operatorname{Enc}(\mathcal{X}) \times \operatorname{Enc}(\mathcal{Y})$ so that $xy \in \operatorname{Enc}(\mathcal{R})$. By Lemma 4.7, $\operatorname{Enc}(\mathcal{G}(\mathcal{Y}, \mathcal{R}, \mathcal{X})) \supset \{\tilde{x} \in \operatorname{Enc}(\mathcal{X}) : \exists \tilde{y} \in \operatorname{Enc}(\mathcal{Y}), z \in \operatorname{Enc}(\mathcal{R}) : \tilde{x}\tilde{y} = z\}$, and hence $x \in \operatorname{Enc}(\mathcal{G}(\mathcal{Y}, \mathcal{R}, \mathcal{X}))$. Likewise, $\{\tilde{y} \in \operatorname{Enc}(\mathcal{Y}) : \exists \tilde{x} \in \operatorname{Enc}(\mathcal{G}(\mathcal{Y}, \mathcal{R}, \mathcal{X})), z \in \operatorname{Enc}(\mathcal{R}) : \tilde{x}\tilde{y} = z\} \subset \operatorname{Enc}(\mathcal{G}(\mathcal{G}(\mathcal{Y}, \mathcal{R}, \mathcal{X}), \mathcal{R}, \mathcal{Y}))$, hence $y \in \operatorname{Enc}(\mathcal{G}(\mathcal{G}(\mathcal{Y}, \mathcal{R}, \mathcal{X}), \mathcal{R}, \mathcal{Y}))$. Thus, $(x, y) \in \operatorname{Enc}(\times^{\text{rev}}((\mathcal{X}, \mathcal{Y}), \mathcal{R}))$ and (4.3) holds. □

Note that $\times^{\text{rev}}((\mathcal{X}, \mathcal{Y}), \mathcal{R}) = (\mathcal{G}(\mathcal{G}(\mathcal{X}, \mathcal{R}, \mathcal{Y}), \mathcal{R}, \mathcal{X}), \mathcal{G}(\mathcal{X}, \mathcal{R}, \mathcal{Y}))$ is an alternative reverse McCormick update of $(\times, \mathbb{R}^2, \mathbb{R})$.

4.2.2 Reverse McCormick updates of univariate functions

Lemma 4.9. *Let* $B \subset \mathbb{R}$ *and consider an injective continuous function* $(u, B, \mathbb{R}) \in \mathcal{L}$. *Furthermore, assume that* $(u^{-1}, u(B), \mathbb{R}) \in \mathcal{L}$ *where* $u(B)$ *refers to the image of the real-valued function* u. *The function* $(u^{\text{rev}}, \mathbb{M}_\emptyset B \times \bar{\mathbb{MR}}, \mathbb{MR}_\emptyset)$ *defined for all* $\mathcal{X} \in \mathbb{M}_\emptyset B$ *and* $\mathcal{R} \in \bar{\mathbb{MR}}$ *by*

$$u^{\text{rev}}(\mathcal{X}, \mathcal{R}) = u^{-1}(\mathcal{T}) \cap \mathcal{X}$$

where $\mathcal{T} = (R^B \cap u(X^B), \operatorname{Enc}(\mathcal{R} \cap u(\mathcal{X})))$ *is a reverse McCormick update of* $(u, B, \mathbb{R})$.

Proof. Let $\mathcal{X} \in \mathbb{M}_\emptyset B$. Suppose that $\operatorname{Enc}(\mathcal{T}) = \emptyset$. Since $(u, \mathbb{M}_\emptyset B, \mathbb{MR})$ is a relaxation function, there does not exist an $x \in \operatorname{Enc}(\mathcal{X})$ so that $u(x) \in \operatorname{Enc}(\mathcal{R})$. Otherwise, since $(u, B, \mathbb{R})$ is continuous and injective, it is invertible on $u(B)$ and u^{-1} is continuous [145, Thm. 4.17]. Since $(u^{-1}, u(B), \mathbb{R}) \in \mathcal{L}$, $u(x) \in \operatorname{Enc}(\mathcal{T})$ implies $x = u^{-1}(u(x)) \in \operatorname{Enc}(u^{-1}(\mathcal{T}))$. □

Remark 4.2. Lemma 4.9 can be used to define the reverse McCormick update of $-(\cdot)$, $(\cdot)^n$ for odd $n \in \mathbb{N}$, exp, log, $\sqrt{\cdot}$, etc. It is also applicable to $\frac{1}{(\cdot)}$ if B is restricted to either the negative or positive reals.

Lemma 4.10. *Let $n \in \mathbb{N}$ be even. Consider $(u, \mathbb{R}, \mathbb{R}) \in \mathcal{L}$ where $u(x) = x^n$ and assume that $(\sqrt[n]{\cdot}, [0, +\infty), \mathbb{R}) \in \mathcal{L}$. The function $(u^{\text{rev}}, \mathbb{MR}_\emptyset \times \bar{\mathbb{MR}}, \mathbb{MR}_\emptyset)$ defined for all $\mathcal{X} \in \mathbb{MR}_\emptyset$ and $\mathcal{R} \in \bar{\mathbb{MR}}$ by*

$$u^{\text{rev}}(\mathcal{X}, \mathcal{R}) = \begin{cases} \emptyset & \text{if } R^B \cap u(X^B) = \emptyset, \\ \sqrt[n]{\mathcal{T}} \cap \mathcal{X} & \text{if } \underline{x} \geq 0, \\ -\sqrt[n]{\mathcal{T}} \cap \mathcal{X} & \text{if } \overline{x} \leq 0, \\ \left(u^{\text{rev}}(X^B, T^B), \left[-\sqrt[n]{\hat{t}}, \sqrt[n]{\hat{t}}\right] \cap u^{\text{rev}}(X^B, T^B)\right) \cap \mathcal{X} & \text{otherwise,} \end{cases}$$

where $\mathcal{T} = (R^B \cap u(X^B) \cap [0, +\infty), \text{Enc}(\mathcal{R}) \cap \text{Enc}(u(\mathcal{X})) \cap [0, +\infty))$ is a reverse McCormick update of $(u, \mathbb{R}, \mathbb{R})$.

Proof. Let $\mathcal{X} \in \mathbb{MR}_\emptyset$. Suppose that $R^B \cap u(X^B) = \emptyset$. Since $(u, \mathbb{IR}_\emptyset, \mathbb{IR})$ is an inclusion function, there does not exist an $x \in X^B$ so that $u(x) \in R^B$.

In the following, assume that $R^B \cap u(X^B) \neq \emptyset$. Note that intersecting $\mathcal{R}$ with the non-negative half space only ensures that no domain violation occurs. Let $\tilde{x} \in \text{Enc}(\mathcal{X})$ so that $u(\tilde{x}) \in \text{Enc}(\mathcal{R})\}$. If $\underline{x} \geq 0$ then $\tilde{x} \geq 0$. By definition of the relaxation function of $\sqrt[n]{\cdot}$, it follows that $\{x \in \mathbb{R} : x \geq 0 \wedge u(x) \in \text{Enc}(\mathcal{R})\} \subset \text{Enc}(\sqrt[n]{\mathcal{T}})$. Similarly, if $\overline{x} \leq 0$ then $\tilde{x} \leq 0$. Since $u(-\tilde{x}) = u(\tilde{x}) \geq 0$ and $u(-\tilde{x}) \in \text{Enc}(\mathcal{R})$, $u(-\tilde{x}) \in \text{Enc}(\mathcal{T})$ so that $\tilde{x} = -(-\tilde{x}) = -\sqrt[n]{u(-\tilde{x})} \in \text{Enc}(-\sqrt[n]{\mathcal{T}})$. Hence, $\{x \in \mathbb{R} : x \leq 0 \wedge u(x) \in \text{Enc}(\mathcal{R})\} \subset \text{Enc}(-\sqrt[n]{\mathcal{T}})$. Otherwise, if $0 \notin X^B$, it is easy to see that

$$\{x \in \mathbb{R} : u(x) \in \text{Enc}(\mathcal{R})\} = \{x \in \mathbb{R} : \exists y \in \text{Enc}(\mathcal{R}), x = -\sqrt[n]{y} \vee x = \sqrt[n]{y}\} \subset \left[-\sqrt[n]{\hat{t}}, \sqrt[n]{\hat{t}}\right].$$

Intersecting with the reverse interval update does not discard any $\tilde{x}$ for which $u(\tilde{x}) \in \text{Enc}(\mathcal{R})$ holds. □

Note that a similar construction is possible to find the reverse McCormick update of the absolute value function.

4.2.3 Inclusion monotonicity of the reverse McCormick updates

Next, it will be shown that reverse McCormick updates are inclusion monotonic while the next subsection focuses on establishing coherent concavity. Note that [155, Lemma 2.4.15] will be referenced multiple times hereafter to establish inclusion monotonicity of a finite composition of inclusion monotonic functions. Though coherent concavity was also assumed in that result, it is not necessary in order to establish inclusion monotonicity of a finite composition of inclusion monotonic functions.

First, note that the intersection update is inclusion monotonic.

Lemma 4.11. *The mapping* $\cap : \mathbb{M}\bar{\mathbb{I}}\mathbb{R} \times \mathbb{M}\bar{\mathbb{I}}\mathbb{R} \to \mathbb{M}\bar{\mathbb{I}}\mathbb{R}$ *defined by* $\cap(\mathcal{X}, \mathcal{Y}) = \mathcal{X} \cap \mathcal{Y}$ *for all* $\mathcal{X}, \mathcal{Y} \in \mathbb{M}\bar{\mathbb{I}}\mathbb{R}$ *is inclusion monotonic on* $\mathbb{M}\bar{\mathbb{I}}\mathbb{R} \times \mathbb{M}\bar{\mathbb{I}}\mathbb{R}$.

Proof. Let $\mathcal{X}_1, \mathcal{X}_2, \mathcal{Y}_1, \mathcal{Y}_2 \in \mathbb{M}\bar{\mathbb{I}}\mathbb{R}$. Then, $\mathcal{X}_1 \subset \mathcal{X}_2$ and $\mathcal{Y}_1 \subset \mathcal{Y}_2$ imply $\mathcal{X}_1 \cap \mathcal{Y}_1 \subset \mathcal{X}_2 \cap \mathcal{Y}_2$. □

Next, the binary operations are considered.

Lemma 4.12. $(+^{\text{rev}}, \mathbb{M}\mathbb{R}^2_{\emptyset} \times \mathbb{M}\bar{\mathbb{I}}\mathbb{R}, \mathbb{M}\mathbb{R}_{\emptyset})$ *is inclusion monotonic on* $\mathbb{M}\mathbb{R}^2_{\emptyset} \times \mathbb{M}\bar{\mathbb{I}}\mathbb{R}$.

Proof. This follows immediately since the negative univariate function [155, Theorem 2.4.29 together with Section 2.8], addition [155, Theorem 2.4.20], the intersection operator (Lemma 4.11) as well as finite composition of inclusion monotonic mappings [155, cf. Lemma 2.4.15] are inclusion monotonic. □

It is helpful to study the extended Gauss-Seidel operator prior to looking at the reverse update of multiplication.

Lemma 4.13. $\mathcal{G}$ *is inclusion monotonic on* $\mathbb{M}\mathbb{R}_{\emptyset} \times \mathbb{M}\bar{\mathbb{I}}\mathbb{R} \times \mathbb{M}\mathbb{R}_{\emptyset}$.

Proof. It was already shown that multiplication [155, Theorem 2.4.23] and the reciprocal function [155, Theorem 2.4.29 together with Section 2.8] as well as finite composition [155, Lemma 2.4.15] are inclusion monotonic. Also note that Γ is inclusion monotonic [127, 4.3.2]. Let $(\mathcal{X}_1, \mathcal{R}_1, \mathcal{Y}_1), (\mathcal{X}_2, \mathcal{R}_2, \mathcal{Y}_2) \in \mathbb{M}\mathbb{R}_{\emptyset} \times \mathbb{M}\bar{\mathbb{I}}\mathbb{R} \times \mathbb{M}\mathbb{R}_{\emptyset}$ so that $(\mathcal{X}_1, \mathcal{R}_1, \mathcal{Y}_1) \subset (\mathcal{X}_2, \mathcal{R}_2, \mathcal{Y}_2)$. If $0 \notin X_2^B$ then $(\mathcal{R}_1' \times \frac{1}{\mathcal{X}_1'})^C \cap (Y_1')^C \subset (\mathcal{R}_2' \times \frac{1}{\mathcal{X}_2'})^C \cap (Y_2')^C$. Otherwise, if $0 \notin X_1^B$ then $(\mathcal{R}_1' \times \frac{1}{\mathcal{X}_1'})^C \cap (Y_1')^C \subset \Gamma(X_1^B, R_1^B, Y_1^B) \cap (Y_1')^C \subset \Gamma(X_2^B, R_2^B, Y_2^B) \cap (Y_2')^C$. Otherwise, if $0 \in X_1^B$ and $0 \notin R_2^B$ then $\Gamma(X_1^B, R_1^B, Y_1^B) \cap (Y_1')^C \subset \Gamma(X_2^B, R_2^B, Y_2^B) \cap (Y_2')^C$. Otherwise, if $0 \in X_1^B$ and $0 \in R_2^B$ then $\Gamma(X_1^B, R_1^B, Y_1^B) \cap (Y_1')^C \subset (Y_1')^C \subset (Y_2')^C$ Thus, $\mathcal{G}$ is inclusion monotonic. □

Lemma 4.14. $(\times^{\text{rev}}, \mathbb{M}\mathbb{R}^2_{\emptyset} \times \mathbb{M}\bar{\mathbb{I}}\mathbb{R}, \mathbb{M}\mathbb{R}_{\emptyset})$ *is inclusion monotonic on* $\mathbb{M}\mathbb{R}_{\emptyset} \times \mathbb{M}\mathbb{R}_{\emptyset} \times \mathbb{M}\bar{\mathbb{I}}\mathbb{R}$.

Proof. Since $\mathcal{G}$ and finite composition [155, Lemma 2.4.15] are inclusion monotonic, the result is immediate. □

Next, the reverse updates of univariate functions are considered.

Lemma 4.15. *Let* $B \subset \mathbb{R}$ *and consider an injective continuous function* $(u, B, \mathbb{R}) \in \mathcal{L}$. *Assume that* $(u^{-1}, u(B), \mathbb{R}) \in \mathcal{L}$. *Then,* u^{rev} *as defined in Lemma 4.9 is inclusion monotonic on* $\mathbb{M}_{\emptyset}B \times \mathbb{M}\bar{\mathbb{I}}\mathbb{R}$.

Proof. Since $(u^{-1}, u(B), \mathbb{R}) \in \mathcal{L}$, it follows that u^{-1} is inclusion monotonic [155, Theorem 2.4.29]. □

Lemma 4.16. *Let* $n \in \mathbb{N}$ *be even. Consider* $(u, \mathbb{R}, \mathbb{R}) \in \mathcal{L}$ *where* $u(x) = x^n$. *Assume that* $(\sqrt[n]{\cdot}, [0, +\infty), \mathbb{R}) \in \mathcal{L}$. *Suppose that the convex and concave envelopes of* $\sqrt[n]{\cdot}$ *are used in calculating relaxations. Then,* u^{rev} *as defined in Lemma 4.10 is inclusion monotonic on* $\mathbb{M}\mathbb{R}_{\emptyset} \times \mathbb{M}\bar{\mathbb{I}}\mathbb{R}$.

Proof. Note that the relaxation function of $\sqrt[n]{\cdot}$ and of the negative operator, the intersection operator and finite composition is inclusion monotonic. Note that $\mathcal{T}$ is inclusion monotonic by construction and so is $u^{\text{rev}}(X^B, T^B)$. Let $(\mathcal{X}_1, \mathcal{R}_1), (\mathcal{X}_2, \mathcal{R}_2) \in \mathbb{M}_\emptyset B \times \bar{\mathbb{M}}\mathbb{R}$ so that $(\mathcal{X}_1, \mathcal{R}_1) \subset (\mathcal{X}_2, \mathcal{R}_2)$. If $R_1^B \cap u(X_1^B) = \emptyset$ or if $\underline{x}_2 \geq 0$ or $\overline{x}_2 \leq 0$ then $u^{\text{rev}}(\mathcal{X}_1, \mathcal{R}_1) \subset u^{\text{rev}}(\mathcal{X}_2, \mathcal{R}_2)$. Otherwise, suppose $\underline{x}_1 \geq 0$. Then $\sqrt[n]{\mathcal{T}_1} \cap \mathcal{X}_1 \subset (u^{\text{rev}}(X_1^B, T_1^B), [-\sqrt[n]{\hat{t}_1}, \sqrt[n]{\hat{t}_1}] \cap u^{\text{rev}}(X_1^B, T_1^B)) \cap \mathcal{X}_1 \subset (u^{\text{rev}}(X_2^B, T_2^B), [-\sqrt[n]{\hat{t}_2}, \sqrt[n]{\hat{t}_2}] \cap u^{\text{rev}}(X_2^B, T_2^B)) \cap \mathcal{X}_2$. A similar argument applies when $\overline{x}_1 \leq 0$. In any other case, inclusion monotonicity follows directly from the properties referenced above and the monotonicity of $\sqrt[n]{\cdot}$, i.e., $\hat{t}_1 \leq \hat{t}_2$ implies that $[-\sqrt[n]{\hat{t}_1}, \sqrt[n]{\hat{t}_1}] \subset [-\sqrt[n]{\hat{t}_2}, \sqrt[n]{\hat{t}_2}]$. □

4.2.4 Coherent concavity of the reverse McCormick updates

Next, it will be shown that reverse McCormick updates are coherently concave. Note that if either $\text{Enc}(\mathcal{F}(\mathcal{X}_1)) = \emptyset$ or $\text{Enc}(\mathcal{F}(\mathcal{X}_2)) = \emptyset$, then the subset condition for coherent concavity holds trivially. Thus, in the proofs below, this case is never considered explicitly.

First, note that the intersection update is coherently concave.

Lemma 4.17. *The mapping* $\cap : \bar{\mathbb{M}}\mathbb{R} \times \bar{\mathbb{M}}\mathbb{R} \to \bar{\mathbb{M}}\mathbb{R}$ *defined by* $\cap(\mathcal{X}, \mathcal{Y}) = \mathcal{X} \cap \mathcal{Y}$ *for all* $\mathcal{X}, \mathcal{Y} \in \bar{\mathbb{M}}\mathbb{R}$ *is coherently concave on* $\bar{\mathbb{M}}\mathbb{R} \times \bar{\mathbb{M}}\mathbb{R}$.

Proof. Suppose $\mathcal{X}_1, \mathcal{X}_2 \in \bar{\mathbb{M}}\mathbb{R}$ and $\mathcal{Y}_1, \mathcal{Y}_2 \in \bar{\mathbb{M}}\mathbb{R}$ are coherent. Let $\lambda \in [0, 1]$. Since $X_1^B = X_2^B$ and $Y_1^B = Y_2^B$, it follows that $X_1^B \cap Y_1^B = X_2^B \cap Y_2^B = (\lambda X_1^B + (1-\lambda)X_2^B) \cap (\lambda Y_1^B + (1-\lambda)Y_2^B)$. Thus, $\cap(\mathcal{X}_1, \mathcal{Y}_1)$ and $\cap(\mathcal{X}_2, \mathcal{Y}_2)$ are coherent.

We will show that $\cap(\text{Conv}(\lambda, (\mathcal{X}_1, \mathcal{Y}_1), (\mathcal{X}_2, \mathcal{Y}_2))) \supset \text{Conv}(\lambda, \cap(\mathcal{X}_1, \mathcal{Y}_1), \cap(\mathcal{X}_2, \mathcal{Y}_2))$. Let $\mathbf{z}_1 \in X_1^C \cap Y_1^C$ and $\mathbf{z}_2 \in X_2^C \cap Y_2^C$. Denote $\mathbf{z} = \lambda \mathbf{z}_1 + (1-\lambda)\mathbf{z}_2$. By construction, $\mathbf{z}_1 \in X_1^C$, $\mathbf{z}_1 \in Y_1^C$ and $\mathbf{z}_2 \in X_2^C$, $\mathbf{z}_2 \in Y_2^C$ so that $\mathbf{z} \in \lambda X_1^C + (1-\lambda)X_2^C$ and $\mathbf{z} \in \lambda Y_1^C + (1-\lambda)Y_2^C$. Thus, $\mathbf{z} \in (\lambda X_1^C + (1-\lambda)X_2^C) \cap (\lambda Y_1^C + (1-\lambda)Y_2^C)$ so that $\lambda(X_1^C \cap Y_1^C) + (1-\lambda)(X_2^C \cap Y_2^C) \subset ((\lambda X_1^C + (1-\lambda)X_2^C) \cap (\lambda Y_1^C + (1-\lambda)Y_2^C))$. □

In particular, note that the proof indicates that $\cap(\mathcal{X}_1, \mathcal{Y}_1) \neq \emptyset$ and $\cap(\mathcal{X}_2, \mathcal{Y}_2) \neq \emptyset$ imply that $\cap(\text{Conv}(\lambda, (\mathcal{X}_1, \mathcal{Y}_1), (\mathcal{X}_2, \mathcal{Y}_2))) \neq \emptyset$.

Next, the binary operations are considered.

Lemma 4.18. $(+^{\text{rev}}, \mathbb{MR}_\emptyset^2 \times \bar{\mathbb{M}}\mathbb{R}, \mathbb{MR}_\emptyset)$ *is coherently concave on* $\mathbb{MR}_\emptyset^2 \times \bar{\mathbb{M}}\mathbb{R}$.

Proof. Note that negative univariate function [155, Theorems 2.4.29 and 2.4.30 together with Section 2.8], addition [155, Theorem 2.4.20] and the intersection operator (Lemmas 4.11 and 4.17) are inclusion monotonic and coherently concave, $(+^{\text{rev}}, \mathbb{IR}_\emptyset^2 \times \bar{\mathbb{IR}}, \mathbb{IR}_\emptyset)$ is inclusion monotonic and finite composition [155, Lemma 2.4.15] is coherently concave. Thus, the result follows.

□

It is helpful to study the extended Gauss-Seidel operator prior to looking at the reverse update of multiplication.

Lemma 4.19. $\mathcal{G}$ *is coherently concave on* $\mathbb{MR}_\emptyset \times \bar{\mathbb{M}}\mathbb{R} \times \mathbb{MR}_\emptyset$.

Proof. Let $\mathcal{X}_1, \mathcal{X}_2 \in \mathbb{MR}_\emptyset$, $\mathcal{R}_1, \mathcal{R}_2 \in \bar{\mathbb{MR}}$ and $\mathcal{Y}_1, \mathcal{Y}_2 \in \mathbb{MR}_\emptyset$ be coherent. Since $X_1^B = X_2^B$, $Y_1^B = Y_2^B$ and $R_1^B = R_2^B$, it follows that $\Gamma(X_1^B, R_1^B, Y_1^B) = \Gamma(X_2^B, R_2^B, Y_2^B)$ so that $\mathscr{G}(\mathcal{X}_1, \mathcal{R}_1, \mathcal{Y}_1)$ and $\mathscr{G}(\mathcal{X}_2, \mathcal{R}_2, \mathcal{Y}_2)$ are coherent.

Suppose $0 \notin X_1^B = X_2^B$. It was already shown that multiplication [155, Theorems 2.4.23 and 2.4.24] and the reciprocal function [155, Theorems 2.4.29 and 2.4.30 together with Section 2.8] are inclusion monotonic and coherently concave and finite compositions of inclusion monotonic, coherently concave functions are coherently concave [155, Lemma 2.4.15]. Also, Lemmas 4.11 and 4.17 show that the intersection operation is coherently concave and inclusion monotonic. Thus, $\mathscr{G}$ is coherently concave in this case.

Next, suppose $0 \in X_1^B$ and $\underline{r}_1 = \underline{r}_2 > 0$ or $\bar{r}_1 = \bar{r}_2 < 0$. Pick $\lambda \in [0,1]$ and let $z_1 \in \Gamma(X_1^B, R_1^B, Y_1^B) \cap Y_1^C$, $z_2 \in \Gamma(X_2^B, R_2^B, Y_2^B) \cap Y_2^C$. Consider $z = \lambda z_1 + (1-\lambda) z_2$. Since $z_1 \in Y_1^C$ and $z_2 \in Y_2^C$, $z \in \lambda Y_1^C + (1-\lambda) Y_2^C$. Note that $z \in \Gamma(X_1^B, R_1^B, Y_1^B) = \Gamma(X_2^B, R_2^B, Y_2^B)$. Thus, $z \in \lambda(\Gamma(X_1^B, R_1^B, Y_1^B) \cap Y_1^C) + (1-\lambda)(\Gamma(X_2^B, R_2^B, Y_2^B) \cap Y_2^C)$ and $\mathscr{G}$ is coherently concave in this case.

In the last case, coherent concavity is immediate. □

Lemma 4.20. $(\times^{\text{rev}}, \mathbb{MR}_\emptyset^2 \times \bar{\mathbb{MR}}, \mathbb{MR}_\emptyset)$ *is coherently concave on* $\mathbb{MR}_\emptyset \times \mathbb{MR}_\emptyset \times \bar{\mathbb{MR}}$.

Proof. Since $\mathscr{G}$ is inclusion monotonic and coherently concave and finite compositions of inclusion monotonic, coherently concave functions are coherently concave [155, Lemma 2.4.15], the result is immediate. □

Next, the reverse updates of univariate functions are considered.

Lemma 4.21. *Let* $B \subset \mathbb{R}$ *and consider an injective continuous function* $(u, B, \mathbb{R}) \in \mathcal{L}$. *Assume that* $(u^{-1}, u(B), \mathbb{R}) \in \mathcal{L}$. *Then,* u^{rev} *as defined in Lemma 4.9 is coherently concave on* $\mathbb{M}_\emptyset B \times \bar{\mathbb{MR}}$.

Proof. Let $\mathcal{X}_1, \mathcal{X}_2 \in \mathbb{MR}_\emptyset$ and $\mathcal{R}_1, \mathcal{R}_2 \in \bar{\mathbb{MR}}$ be coherent, i.e., $X_1^B = X_2^B$ and $R_1^B = R_2^B$. Note that $u^{\text{rev}}(\mathcal{X}_1, \mathcal{R}_1)$ and $u^{\text{rev}}(\mathcal{X}_2, \mathcal{R}_2)$ are coherent. Since $(u^{-1}, u(B), \mathbb{R}) \in \mathcal{L}$, it follows that u^{-1} is coherently concave [155, Theorem 2.4.30]. □

Lemma 4.22. *Let* $n \in \mathbb{N}$ *be even. Consider* $(u, \mathbb{R}, \mathbb{R}) \in \mathcal{L}$ *where* $u(x) = x^n$. *Assume that* $(\sqrt[n]{\cdot}, [0, +\infty), \mathbb{R}) \in \mathcal{L}$. *Then,* u^{rev} *as defined in Lemma 4.10 is coherently concave on* $\mathbb{MR}_\emptyset \times \bar{\mathbb{MR}}$.

Proof. Let $\mathcal{X}_1, \mathcal{X}_2 \in \mathbb{MR}_\emptyset$ and $\mathcal{R}_1, \mathcal{R}_2 \in \bar{\mathbb{MR}}$ be coherent, i.e., $X_1^B = X_2^B$ and $R_1^B = R_2^B$. Note that $u^{\text{rev}}(\mathcal{X}_1, \mathcal{R}_1)$ and $u^{\text{rev}}(\mathcal{X}_2, \mathcal{R}_2)$ are coherent.

By assumption, the relaxation function of $\sqrt[n]{\cdot}$ is coherently concave. Likewise, $-\sqrt[n]{\cdot}$ is coherently concave which follows from coherent concavity of the negative operator and the composition theorem [155, Lemma 2.4.15]. It follows that $\sqrt[n]{\cdot}$ and $-\sqrt[n]{\cdot}$ are relaxation functions. As the intersection operator is coherently concave (Lemma 4.17), coherent concavity for the cases $\underline{x} \geq 0$ and $\bar{x} \leq 0$ follows. Otherwise, we must consider two potential roots. Define $(\tilde{u}^{-1}, [0, +\infty), \mathbb{P}(\mathbb{R}))$ for each $x \in [0, +\infty)$ by $\tilde{u}^{-1}(x) = \{-\sqrt[n]{x}, \sqrt[n]{x}\}$ and note that $u(y) = x$ for each $y \in \tilde{u}^{-1}(x)$ and $x \in \tilde{u}^{-1}(u(x))$. It is easy to see that $-\sqrt[n]{\cdot}$ and $\sqrt[n]{\cdot}$ are the convex and concave envelopes of $\tilde{u}^{-1}$ so that we can use the construction of the relaxation function of $\tilde{u}^{-1}$ in Eq. (3.2). Furthermore, $t^{\min}(T^B) = t^{\max}(T^B) = \bar{t}$ and

$\bar{t} \geq \hat{t}$ in this case so that $\operatorname{mid}(\underset{\smile}{t}, \hat{t}, t^{\min}(T^B)) = \operatorname{mid}(\underset{\smile}{t}, \hat{t}, t^{\max}(T^B)) = \hat{t}$. This is equivalent to the relaxation we obtain by using Equation (3.2). It has already been established that Equation (3.2) provides for a coherently concave relaxation function [155, Theorem 2.4.30] so that, together with Lemma 4.17, coherent concavity of the last case follows. □

4.3 Using reverse McCormick propagation in CSPs and in global optimization

Consider a CSP with variables $\mathbf{y} = (y_1, \ldots, y_n)$, domains $D \in \mathbb{IR}^n$ and constraints

$$\mathbf{g}(\mathbf{y}) \leq \mathbf{0}, \tag{4.5}$$
$$\mathbf{h}(\mathbf{y}) = \mathbf{0}, \tag{4.6}$$

where $\mathbf{g} : D \to \mathbb{R}^{n_g}$ and $\mathbf{h} : D \to \mathbb{R}^{n_h}$ are $\mathcal{L}$-factorable functions.

Suppose that the variables $\mathbf{y} \in D$ can be partitioned into *independent* and *dependent* variables, $\mathbf{p} \in P \in \mathbb{IR}^{n-m}$ and $\mathbf{z} \in X \in \mathbb{IR}^m$, respectively, where $P \times X = D$. Consider the set-valued map $\mathbf{x} : P \to \mathbb{P}(X)$ defined by: $\mathbf{p} \mapsto \{\boldsymbol{\xi} \in X : \mathbf{g}(\boldsymbol{\xi}, \mathbf{p}) \leq \mathbf{0}, \mathbf{h}(\boldsymbol{\xi}, \mathbf{p}) = \mathbf{0}\}$. In words, this mapping returns for each $\mathbf{p} \in P$ all points in X that are feasible in the constraints (4.5) and (4.6) and thus are solutions of the CSP.

Remark 4.3. It is *not* assumed that $m = n_h$. The proposed method will work for any choice of m. In particular note that is often not possible to find a closed form for $\mathbf{x}$ nor is nonempty $\mathbf{x}(\mathbf{p})$ or $\mathbf{x}(\mathbf{p})$ a singleton immediate in many cases.

In this section, we will first discuss how reverse McCormick propagation can be applied to utilize equality and inequality constraints. Next, we will compare different full-space and reduced-space relaxations of nonlinear programs and we will conclude with a discussion on how to partition the variables into independent and dependent ones.

4.3.1 Solving CSPs with equality and inequality constraints

For easier notation, define $\mathbf{c} : D \to \mathbb{R}^{n_g+n_h}$ with $c_i(\mathbf{y}) = g_i(\mathbf{y})$ for $i = 1, \ldots, n_g$ and $c_{i+n_g}(\mathbf{y}) = h_i(\mathbf{y})$ for $i = 1, \ldots, n_h$ and introduce $\mathcal{N} \in \mathbb{MIR}^{n_g+n_h}$ with $\mathcal{N}_i = ((-\infty, 0], (-\infty, 0])$, $i = 1, \ldots, n_g$ and $\mathcal{N}_{i+n_g} = ([0, 0], [0, 0])$, $i = 1, \ldots, n_h$. Let $\mathcal{Y}^0 : P \to \mathbb{MIR}^n$ where $\mathcal{Y}^0_i = (X_i, [\underline{x}_i, \overline{x}_i])$ for $i = 1, \ldots, m$ and $\mathcal{Y}^0_{i+m} = (P_i, [p_i, p_i])$ for $i = 1, \ldots, n - m$.

$\mathcal{Y}^0$ can be interpreted as an a priori enclosure of the solution set of the CSP when $y_{i+m} = p_i$, $i = 1, \ldots, n - m$. Using the idea of constraint propagation on the DAG of $\mathbf{c}$, several avenues to tighten $\mathcal{Y}^0$ exist. First, it is possible to discard parts of D for which it can be guaranteed that no $\mathbf{y}$ exists that satisfies Equations (4.5) and (4.6). Most easily, this can be achieved by reverse interval propagation [174], which considers the bounds only. Second, reverse McCormick propagation provides a means to improve the original bounds and relaxations to find new bounds and relaxations that are at least as tight as the original relaxations and possibly nonconstant.

Let $(\mathcal{S}, \pi_o)$ be a $\mathcal{L}$-computational sequence corresponding to $\mathbf{c}$. Recall the definition of $\mathbf{c}_{\mathcal{S}}^{\text{rev}}$, cf. Equation (4.2), and note that for each $\mathbf{p} \in P$ and $\boldsymbol{\xi} \in \mathbf{x}(\mathbf{p})$ there exists a $\mathbf{n} \in \text{Enc}(\mathcal{N})$ so that $\mathbf{c}_{\mathcal{S}}^{\text{rev}}((\boldsymbol{\xi}, \mathbf{p}), \mathbf{n}) = (\boldsymbol{\xi}, \mathbf{p})$. Consider the reverse McCormick propagation of $\mathbf{c}$:

$$\mathcal{C}_{\mathcal{S}}^{\text{rev}}(((X, X), (P, [\mathbf{p}, \mathbf{p}])), \mathcal{N}) \equiv ((\tilde{X}, [\check{\mathbf{x}}(\mathbf{p}), \hat{\mathbf{x}}(\mathbf{p})]), (\tilde{P}, [\check{\mathbf{p}}(\mathbf{p}), \hat{\mathbf{p}}(\mathbf{p})])). \tag{4.7}$$

Note that $\mathcal{C}_{\mathcal{S}}^{\text{rev}}$ is a relaxation function of $\mathbf{c}_{\mathcal{S}}^{\text{rev}}$ by Theorem 4.4. As the following theorem shows, one interpretation of Equation 4.7 is that it defines $\check{\mathbf{x}}, \hat{\mathbf{x}} : P \to \mathbb{R}^m$, which are convex and concave relaxations of $\mathbf{x}$ on P, respectively, (and, less interestingly, $\check{\mathbf{p}}, \hat{\mathbf{p}} : P \to \mathbb{R}^{n-m}$, which are convex and concave relaxations of the identity function $\mathbf{p}$ on P).

Theorem 4.5. *Consider $\mathcal{C}_{\mathcal{S}}^{\text{rev}}$, a relaxation function of $\mathbf{c}_{\mathcal{S}}^{\text{rev}}$ on $(X \times P) \times \bar{\mathbb{R}}^{n_g+n_h}$. Let $\check{\mathbf{x}}, \hat{\mathbf{x}} : P \to \mathbb{R}^m$ be as defined by Equation 4.7. Then, $\check{\mathbf{x}}, \hat{\mathbf{x}}$ are convex and concave relaxations of $\mathbf{x}$ on P, respectively.*

Proof. Let $\mathbf{x} \in X$, $\mathbf{p} \in P$ and $\boldsymbol{\phi} \in \text{Enc}(\mathcal{N})$. Note that $\mathbf{c}_{\mathcal{S}}^{\text{rev}}((\mathbf{x}, \mathbf{p}), \phi) = (\mathbf{x}, \mathbf{p})$ if $\mathbf{c}_{\mathcal{S}}(\mathbf{x}, \mathbf{p}) = \boldsymbol{\phi}$. Since $\mathcal{C}_{\mathcal{S}}^{\text{rev}}$ is a relaxation function of $\mathbf{c}_{\mathcal{S}}^{\text{rev}}$, it follows for such $(\mathbf{x}, \mathbf{p})$ that

$$\text{Enc}(\mathcal{C}_{\mathcal{S}}^{\text{rev}}(((X, X), (P, [\mathbf{p}, \mathbf{p}])), \mathcal{N})) \supset \text{Enc}(\mathcal{C}_{\mathcal{S}}^{\text{rev}}(((X, [\mathbf{x}, \mathbf{x}]), (P, [\mathbf{p}, \mathbf{p}])), \mathcal{N})) \supset ([\mathbf{x}, \mathbf{x}], [\mathbf{p}, \mathbf{p}]).$$

In particular, $\check{\mathbf{x}}(\mathbf{p}) \leq \inf\{\mathbf{x}(\mathbf{p})\} \leq \sup\{\mathbf{x}(\mathbf{p})\} \leq \hat{\mathbf{x}}(\mathbf{p})$.

Pick $\mathbf{p}_1, \mathbf{p}_2 \in P$ and $\lambda \in (0, 1)$. Consider $\mathcal{Y}_1 = ((X, X), (P, [\mathbf{p}_1, \mathbf{p}_1])) \times \mathcal{N}$ and $\mathcal{Y}_2 = ((X, X), (P, [\mathbf{p}_2, \mathbf{p}_2])) \times \mathcal{N}$. Since $\mathcal{C}_{\mathcal{S}}^{\text{rev}}$ is coherently concave on $(X \times P) \times \bar{\mathbb{R}}^{n_g+n_h}$, it follows that

$$\mathcal{C}_{\mathcal{S}}^{\text{rev}}(\text{Conv}(\lambda, \mathcal{Y}_1, \mathcal{Y}_2)) \supset \text{Conv}(\lambda, \mathcal{C}_{\mathcal{S}}^{\text{rev}}(\mathcal{Y}_1), \mathcal{C}_{\mathcal{S}}^{\text{rev}}(\mathcal{Y}_2)),$$

which implies that

$$\begin{aligned} \check{\mathbf{x}}(\lambda\mathbf{p}_1 + (1-\lambda)\mathbf{p}_2) &\leq \lambda\check{\mathbf{x}}(\mathbf{p}_1) + (1-\lambda)\check{\mathbf{x}}(\mathbf{p}_2) \\ \hat{\mathbf{x}}(\lambda\mathbf{p}_1 + (1-\lambda)\mathbf{p}_2) &\geq \lambda\hat{\mathbf{x}}(\mathbf{p}_1) + (1-\lambda)\hat{\mathbf{x}}(\mathbf{p}_2). \end{aligned}$$

Thus, $\check{\mathbf{x}}, \hat{\mathbf{x}}$ are convex and concave relaxations of $\mathbf{x}$ on P, respectively. □

In other words, given $\mathbf{p} \in P$ and $\boldsymbol{\xi} \in \mathbf{x}(\mathbf{p})$, it holds that $\boldsymbol{\xi} \in [\check{\mathbf{x}}(\mathbf{p}), \hat{\mathbf{x}}(\mathbf{p})]$. Also note that a particular possible outcome of the reverse McCormick propagation is

$$\mathcal{C}_{\mathcal{S}}^{\text{rev}}(((X, X), (P, [\mathbf{p}, \mathbf{p}])), \mathcal{N}) = ((\tilde{X}, \emptyset), (\tilde{P}, \emptyset))),$$

in which case $\mathbf{x}(\mathbf{p}) = \emptyset$.

The sequence of the calculations for the reverse update $\mathcal{C}_{\mathcal{S}}^{\text{rev}}$ is outlined in Figure 4.2. In contrast to the evaluation of natural McCormick extensions, the forward evaluation of the relaxation functions in Step (1) is initialized differently. The results of this evaluation are interval bounds on $\mathbf{g}(X, P)$ and $\mathbf{h}(X, P)$ as well as a particular kind of relaxations of $\mathbf{g}$ and $\mathbf{h}$ on P, here denoted by $\check{\mathbf{g}}(X, \mathbf{p})$, etc. From the properties of the relaxation function it follows that $\check{\mathbf{g}}(X, \cdot)$ is convex on P and that $\check{g}_i(X, \mathbf{p}) \leq g_i(\mathbf{x}, \mathbf{p})$, $\forall(\mathbf{x}, \mathbf{p}) \in X \times P$ and $i = 1, \ldots, n_g$. Similarly, $\hat{\mathbf{g}}(X, \mathbf{p})$ denotes an analogue concave relaxation of $\mathbf{g}$. In

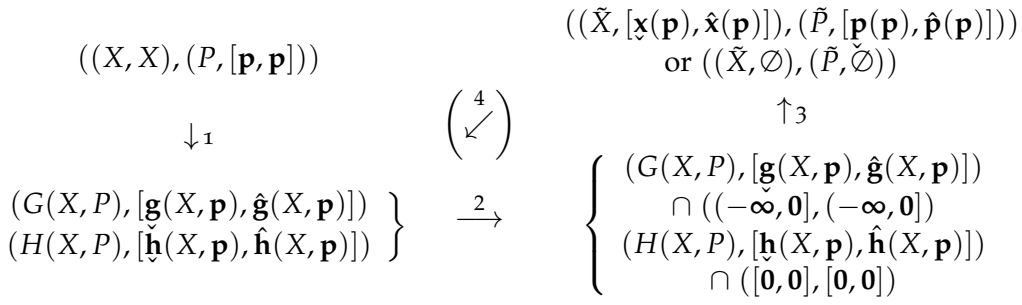


Figure 4.2: Principle of forward-reverse McCormick update to construct relaxations of the implicit set-valued mapping $\mathbf{x}(\cdot)$: forward evaluation of relaxation functions [156] to obtain a particular kind of relaxations of $\mathbf{g}$ and $\mathbf{h}$ on P (1), intersection with constraint information (2), and reverse propagation of additional information (3). This procedure can be iterated on if desired (4).

Step (2), the constraint information is intersected with the relaxation functions of the constraints. This tightens the relaxations without losing the convexity and concavity properties. Step (3) propagates this information back to the variables so that we obtain relaxations of $\mathbf{x}$ evaluated at $\mathbf{p}$ or the information that $\mathbf{x}(\mathbf{p}) = \emptyset$. It is also shown that the procedure can be repeated in order to further improve the computed relaxations (Step (4)).

Let $\mathcal{Y}^{k+1} = \mathcal{C}_{\mathcal{S}}^{\mathrm{rev}}(\mathcal{Y}^k, \mathcal{N})$, $k = 0, 1, \ldots$. Note that the coherent concavity property of $\mathcal{Y}^k$ is guaranteed only for a fixed k so that it is important that the number of reverse updates is equal for all $\mathbf{p} \in P$.

Avoiding domain violations Definition 3.3 ensures that the natural function $\mathbf{f}_{\mathcal{S}}$ of a computational sequence $(\mathcal{S}, \pi_o)$, and, in particular, each participating univariate function, is defined at each point of its natural domain $D_{\mathcal{S}}$ and hence can be safely evaluated there. However, the natural domain of a complicated computational sequence is not easily obtained. If the natural function is evaluated at a point outside its domain, which is possible due to difficulty in practically establishing the exact natural domain, the domain of at least one univariate function will be violated. Definition 3.12 further restricts the natural domains of the natural interval and McCormick extensions. Due to the inherent conservatism of the interval and McCormick techniques, domain violations are also potentially possible for $X \in \mathbb{I}D_{\mathcal{S}}$ or $\mathcal{X} \in \mathbb{M}D_{\mathcal{S}}$. In order to avoid either problem, the following convention is used. Consider $(u, B, \mathbb{R}) \in \mathcal{L}$ and suppose that $B \in \mathbb{I}R$, which is true for many common univariate functions. If $x \notin B$ then set $u(x) = \mathrm{NaN}$. For $X \in \mathbb{I}\mathbb{R}$ or $\mathcal{X} \in \mathbb{M}\mathbb{R}$ with $X \not\subset B$ or $X^B \not\subset B$, the evaluation of $u(X)$ or $u(\mathcal{X})$ is undefined whereas $u(X \cap B)$ or $u(\mathcal{X} \cap (B, B))$ is always defined if our convention $u(\emptyset) = \emptyset$ is used. Given any $X \in \mathbb{I}\mathbb{R}^{n_i}$ or $\mathcal{X} \in \mathbb{M}\mathbb{R}^{n_i}$, this approach continues to construct valid enclosures and

relaxations of $\tilde{\mathbf{f}} : \mathbb{R}^{n_i} \to \mathbb{R}^{n_o}_{\emptyset}$ defined by

$$\tilde{\mathbf{f}}(\mathbf{x}) = \begin{cases} \mathbf{f}_{\mathcal{S}}(\mathbf{x}) & \text{if } \mathbf{x} \in D_{\mathcal{S}}, \\ \text{NaN} & \text{otherwise.} \end{cases}$$

Points outside the natural domain evaluate to NaN and, by our convention, NaN is an element of any interval so that any interval-valued or McCormick-valued function satisfies the inclusion property for such $\mathbf{x}$. On the other hand, the natural interval or McCormick extensions bound or relax the natural function at each point that is contained in the natural domain by its usual properties. Overall, this convention allows us to circumvent difficulties with domain violations without losing the inclusion or relaxation function properties. In particular, it provides more directly useful information than throwing a flag indicating that a domain violation occurred.

4.3.2 Constructing relaxations for reduced-space optimization problems

Consider

$$\begin{aligned} f^* = \min_{\mathbf{z} \in X, \mathbf{p} \in P} \; & f(\mathbf{z}, \mathbf{p}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{z}, \mathbf{p}) \leq \mathbf{0}, \\ & \mathbf{h}(\mathbf{z}, \mathbf{p}) = \mathbf{0} \end{aligned} \tag{P}$$

where $f : X \times P \to \mathbb{R}$, $\mathbf{g} : X \times P \to \mathbb{R}^{n_g}$ and $\mathbf{h} : X \times P \to \mathbb{R}^{n_h}$ are $\mathcal{L}$-factorable.

Define the set-valued mapping $\phi : P \to \mathbb{P}(\mathbb{R})$ for each $\mathbf{p} \in P$ by $\phi(\mathbf{p}) = \{f(\mathbf{z}, \mathbf{p}) : \mathbf{z} \in X, \mathbf{g}(\mathbf{z}, \mathbf{p}) \leq \mathbf{0}, \mathbf{h}(\mathbf{z}, \mathbf{p}) = \mathbf{0}\}$. It is obvious that $f^* = \min_{\mathbf{p} \in P} \inf \phi(\mathbf{p})$.

Let $\tilde{X}$ and $\tilde{P}$ denote the results of a reverse interval update as outlined above and illustrated in Figure 4.2. First, note that $\tilde{X} \times \tilde{P}$ is a superset of the feasible region by construction of the reverse interval update. Recall that the procedure described in the previous section provides valid relaxations of the set-valued mapping $\mathbf{x}$, $\underset{\smile}{\mathbf{x}}$ and $\hat{\mathbf{x}}$. These can be used to calculate generalized relaxation functions of f. To this extent, let $\mathcal{F}$ denote the natural McCormick extension of $\mathbf{f}$ and we will define

$$[\underset{\smile}{\phi}(\mathbf{p}), \hat{\phi}(\mathbf{p})] \equiv (\mathcal{F}((\tilde{X}, [\underset{\smile}{\mathbf{x}}(\mathbf{p}), \hat{\mathbf{x}}(\mathbf{p})]), (\tilde{P}, [\mathbf{p}, \mathbf{p}])))^C.$$

Proposition 4.2. *Consider*

$$\phi^* = \min_{\mathbf{p} \in \tilde{P}} \underset{\smile}{\phi}(\mathbf{p}). \tag{R1}$$

Then, (R1) *is a convex program and* $f^* \geq \phi^*$.

Proof. (R1) is a convex program since $\tilde{P}$ is a convex set and $\underset{\smile}{\phi}$ is convex on $\tilde{P}$. $f^* \geq \phi^*$ follows immediately from $\underset{\smile}{\phi}(\mathbf{p}) \leq \inf \phi(\mathbf{p})$ [155, Theorem 2.7.13]. □

Proposition 4.3. *Let* $\check{f}$, $\check{\mathbf{g}}$ *and* $\check{\mathbf{h}}$ *denote the standard convex McCormick relaxations of* f, $\mathbf{g}$ *and* $\mathbf{h}$, *respectively, on* $\tilde{X} \times \tilde{P}$ *and let* $\hat{\mathbf{h}}$ *denote the standard concave McCormick relaxation of* $\mathbf{h}$ *on* $\tilde{X} \times \tilde{P}$. *Consider*

$$
\begin{aligned}
f_1 = \min_{\mathbf{z}\in\tilde{X},\mathbf{p}\in\tilde{P}} \ & \check{f}(\mathbf{z},\mathbf{p}) \\
\text{s.t.} \ & \check{\mathbf{g}}(\mathbf{z},\mathbf{p}) \leq \mathbf{0}, \\
& \check{\mathbf{h}}(\mathbf{z},\mathbf{p}) \leq \mathbf{0} \leq \hat{\mathbf{h}}(\mathbf{z},\mathbf{p}), \\
& \check{\mathbf{x}}(\mathbf{p}) \leq \mathbf{z} \leq \hat{\mathbf{x}}(\mathbf{p}),
\end{aligned}
\tag{R2}
$$

Then, $f^* \geq f_1 \geq \phi^*$.

Proof. It is clear that (R2) is a relaxation of (P) so that $f^* \geq f_1$. Note that $[\check{f}(\mathbf{z},\mathbf{p}), \hat{f}(\mathbf{z},\mathbf{p})] = (\mathcal{F}((\tilde{X},[\mathbf{z},\mathbf{z}]),(\tilde{P},[\mathbf{p},\mathbf{p}])))^C$ holds for the standard McCormick relaxation of f on $\tilde{X} \times \tilde{P}$. Inclusion monotonicity of the natural McCormick extensions implies that for any $\mathbf{p} \in \tilde{P}$ and $\mathbf{z} \in [\check{\mathbf{x}}(\mathbf{p}), \hat{\mathbf{x}}(\mathbf{p})]$, $\mathcal{F}((\tilde{X},[\mathbf{z},\mathbf{z}]),(\tilde{P},[\mathbf{p},\mathbf{p}])) \subset \mathcal{F}((\tilde{X},[\check{\mathbf{x}}(\mathbf{p}),\hat{\mathbf{x}}(\mathbf{p})]),(\tilde{P},[\mathbf{p},\mathbf{p}]))$ and thus $\check{f}(\mathbf{z},\mathbf{p}) \geq \check{\phi}(\mathbf{p})$ so that $f_1 \geq \phi^*$. □

Remark 4.4. (R1) and (R2) are valid relaxations of (P). It is known that McCormick relaxations can be nonsmooth functions [121]. Thus, while (R2) is a tighter relaxation of (P), it potentially requires the solution of a convex nonsmooth program with nonlinear nonsmooth constraints. While methods to solve such programs have been proposed [e.g., 82, 98, 115], the authors are not aware of robust commercial or freely available algorithms. In order to solve (R2), constraints can be linearized using subgradients [121] to construct an outer-approximation. In this case, the consequence of Proposition 4.3 is no longer guaranteed to hold. On the other hand, convex nonsmooth programs with box-constraints can be solved using the method provided in [113], so that a practical method is available to solve (R1). Furthermore, (R1) requires the solution of a $n - m$-dimensional optimization problem whereas (R2) is n-dimensional.

Remark 4.5. An alternative method to obtain a relaxation of (P) is the auxiliary variable method which introduces additional variables and constraints for each factor that appears in the DAG [159, 165–167]. Its relaxations, prior to linearization, are at least as tight as McCormick relaxations [165, p. 127f] and are differentiable functions. However, the dimension of the resulting nonlinear convex optimization problem is (much) larger. It is typically linearized so that the more robust and more efficient linear programming algorithms can be used. Again, no general comparison of the tightness of different relaxations is possible once the linearization is performed. Also, this approach does not include the constraint $\check{\mathbf{x}}(\mathbf{p}) \leq \mathbf{z} \leq \hat{\mathbf{x}}(\mathbf{p})$ in the relaxation so no direct comparison with (R1) and (R2) in terms of tightness is possible.

Suppose it is known that UBD is a valid upper bound on the optimal objective function value of (P), e.g., there exists a $(\mathbf{z}^\dagger, \mathbf{p}^\dagger)$ feasible in (P) with $f(\mathbf{z}^\dagger, \mathbf{p}^\dagger) = UBD$. Similarly, suppose that LBD is a valid lower bound on the optimal objective function value, e.g.,

there does not exists a $(\mathbf{z}^\dagger, \mathbf{p}^\dagger)$ feasible in (P) with $f(\mathbf{z}^\dagger, \mathbf{p}^\dagger) < LBD$. Both cases are very common in the context of a branch-and-bound algorithm. Consider

$$\begin{aligned} f^\ddagger = \min_{\mathbf{z}\in X, \mathbf{p}\in P} \; & f(\mathbf{z}, \mathbf{p}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{z}, \mathbf{p}) \leq \mathbf{0}, \\ & \mathbf{h}(\mathbf{z}, \mathbf{p}) = \mathbf{0}, \\ & f(\mathbf{z}, \mathbf{p}) - UBD \leq 0, \\ & LBD - f(\mathbf{z}, \mathbf{p}) \leq 0. \end{aligned}$$

It is clear that $f^\ddagger = f^*$ since $(\mathbf{z}^\dagger, \mathbf{p}^\dagger)$ is feasible in (P). However, we can potentially strengthen the relaxations $\check{\mathbf{x}}$, $\hat{\mathbf{x}}$ and thus also ϕ^* or f^1 by including $f(\mathbf{z}, \mathbf{p}) - UBD \leq 0$ and $LBD - f(\mathbf{z}, \mathbf{p}) \leq 0$ in the reverse propagation outlined in Section 4.3.1.

4.3.3 Partitioning variables

A discussion on how to partition the variables into X and P concludes this section. We begin by analyzing the two extreme cases: $m = 0$ and $m = n$.

First consider $m = 0$. Here, $\mathcal{Y}$ is initialized using a point, i.e., $\mathcal{Y}_i^0 = (P_i, [p_i, p_i])$ for each $i = 1, \ldots, n$, constructing the tightest relaxations of $\mathbf{c}(\mathbf{p})$ after the forward evaluation. However, only two outcomes are possible after the reverse propagation, either $\mathcal{Y}_i^1 = (\tilde{P}_i, [p_i, p_i])$ or $\mathcal{Y}_i^1 = (\tilde{P}_i, \emptyset)$. While the latter case indicates that $\mathbf{p}$ violates at least one of the constraints, it is not clear how this information can be exploited numerically. For example, it is not clear how to obtain a hyperplane separating infeasible from potentially feasible points.

Next consider $m = n$. In this case, $\mathcal{Y}$ is initialized using the interval bounds, i.e., $\mathcal{Y}_i^0 = (P_i, P_i)$ for each $i = 1, \ldots, n$. This will yield looser relaxations of $\mathbf{c}$ after the forward evaluation and since $\mathcal{Y}$ is constant, we will obtain $\mathcal{Y}^1 = (\tilde{P}, \check{P})$ after the reverse propagation where $\check{P} \in \mathbb{I}_\emptyset P$ is an *interval*. Actually, in this case the reverse McCormick propagation yields the same information as the reverse interval propagation given that the exact image for each univariate function is used as the interval extension and the envelopes are used as the relaxations.

The advantages of the proposed method over interval methods are obtained for partitions between the two extremes listed above. A partitioning with $m = n_h$ such that there exists a unique implicit function $\mathbf{x} : P \to X$ with $\mathbf{h}(\mathbf{x}(\mathbf{p}), \mathbf{p}) = \mathbf{0}$ for all $\mathbf{p} \in P$ is more favorable. In our numerical experience, this partitioning gave results that were better compared to interval reverse propagation. Interval Newton methods can be used to verify the existence and uniqueness of $\mathbf{x}$, see [127, Ch. 5]. Additional inequality constraints can be used to reduce X and P further.

4.4 Implementation

In this section, an implementation of the reverse interval and McCormick propagation in C++ is presented. First, it is briefly discussed how the DAG of a factorable function can be easily constructed. Next, it is shown how forward and reverse interval and McCormick calculations can be performed on this DAG. Consider a factorable function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$. In this section, *independent* and *dependent* variables will refer to $\mathbf{y}$ and $\mathbf{f}(\mathbf{y})$, respectively. The boost interval library is used for the interval calculations [120] and MC++ provides the necessary routines for McCormick objects [40].

The first step is the parsing of the factorable function to construct the DAG. In C++ this can be easily achieved using function and operator overloading. The DAG is stored as an array. Each element of the array corresponds to one factor of the factorable function including factors for the assignment of independent variables. Each element stores the operation type as well as pointers to its parent element(s), an interval and a McCormick object (as defined by MC++). Optionally, a constant parameter can be stored, which is used to keep track of, for example, constant exponents or factors. While the first n array elements correspond to the independent variables, pointers to the dependent variables must be stored. Note that after the DAG has been constructed, all remaining operations are performed on this DAG object.

Prior to a forward interval/McCormick pass, the interval/McCormick objects of the independent variables are initialized. During the forward pass each factor is visited in sequence and the factor's interval/McCormick object is updated according to the operation type using the pointers to parents' values. After the forward pass, the interval/McCormick objects of the dependent variables store the values, which could have been alternatively calculated using traditional methods.

Prior to a reverse pass, the interval/McCormick objects of the dependent variables are updated based on the information supplied by the constraints. Then, each factor is visited in reverse order. A reverse interval/McCormick update is performed and the parents' interval/McCormick objects are updated accordingly. After the reverse pass, the independent variables now store the updated interval/McCormick values. If during the reverse pass one of the intervals or McCormick objects of a factor is set to the empty set then the calculation can be aborted and the result of the reverse propagation is the empty set.

Note that MC++ also provides functionality to calculate subgradients of the convex and concave relaxations [121]. This functionality is essential when the relaxations are to be used in convex optimization algorithms. The present implementation also provides routines to update the subgradients during the reverse pass accordingly.

Additionally, the implementation allows the user to provide constraints on the domains of intermediate factors. These can avoid domain violations as outlined at the end of Section 4.3.1 and they are already taken into account during the forward interval or McCormick pass.

Lastly, it is possible to generate code automatically, in any programming language, that implements any combination of the discussed computations. Similar to source code

transformation in automatic differentiation [70], the produced code can be executed to efficiently evaluate $\underset{\sim}{\mathbf{x}}(\cdot)$ and $\hat{\mathbf{x}}(\cdot)$, for example.

4.5 Case studies

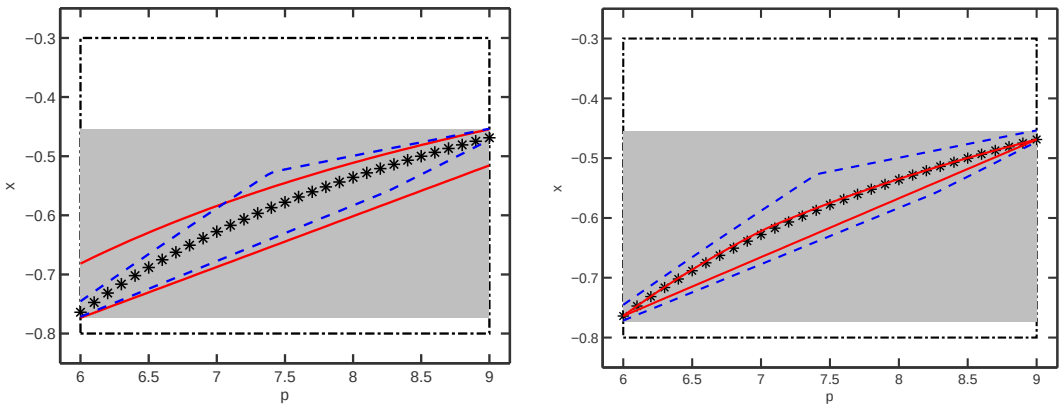
In this section, we will present illustrative case studies that show how enclosures of the solution sets can be obtained from the reverse McCormick propagation and that these compare favorably to the enclosures computed with reverse interval propagation. In the first case study, the constraints define a unique implicit function on P. It is taken from [164] and the results are compared. The second case study compares the feasible region of the relaxed program obtained using reverse McCormick propagation to the feasible region of the standard McCormick relaxation. The third case study focuses on constraints defining a non-unique implicit mapping. The fourth case study shows the effect of a reverse interval propagation pre-processing step when there is no feasible z for some p. The fifth case study shows that relaxations can be sensibly calculated even when there are no feasible z for some p in the interior of P. The sixth case study demonstrates how information from inequality constraints can be incorporated. The last case study illustrates how relaxations of the objective function can be significantly improved by incorporating information from the constraints.

We only consider univariate functions from the library $\mathcal{L} = \{(\cdot)^l, \sqrt[l]{\cdot}, \log, \exp\}$, $l \in \mathbb{N}$. However, the method can be applied to any other univariate functions that satisfy Assumptions 3.1 and 3.3.

4.5.1 Equality constraints

Example 4.1. Let $X = [-0.8, -0.3]$ and $P = [6, 9]$. Consider $h(z, p) = z^2 + zp + 4$ with $(z, p) \in X \times P$. Note that $h(z, p) = 0$ implicitly defines a set-valued mapping $x : P \to \mathbb{P}(X) : p \mapsto \{-\frac{p}{2} + \sqrt{\left(\frac{p}{2}\right)^2 - 4}\}$ so that $h(\xi, p) = 0$ for all $p \in P$ and $\xi \in x(p)$. While Figure 4.3(a) shows the result after one iteration of the reverse McCormick propagation, Figure 4.3(b) depicts the effect of 10 reverse propagation iterations. In both figures the relaxations are compared to those calculated using the method presented in [164]. Note that the calculations for 60 different values of p take a total of 0.0021s, 0.0039s and 0.014s in the case of one reverse propagation, ten reverse propagations and one iteration of the parametric Gauss-Seidel method given in [164] with $\lambda = 0.5$, respectively. Thus, the new method is faster and provides tighter relaxations.

Example 4.2. Let $X = [-3, 5]$ and $P = [-3, 4]$. Consider $h(z, p) = (\sqrt{p + 4} - 3)(\log(p^2 + 1) - z)$ with $(z, p) \in X \times P$. Note that $h(z, p) = 0$ implicitly defines a set-valued mapping $x : P \to \mathbb{P}(X) : p \mapsto \{\log(1 + p^2)\}$ so that $h(\xi, p) = 0$ for all $p \in P$ and $\xi \in x(p)$. The results of a single reverse McCormick propagation are shown in Figure 4.4(a). Additionally, we also show two different relaxations of the non-convex feasible space $\{(z, p) \in X \times P : h(z, p) = 0)\}$ (shown in asterisks). A common way to relax constraints is the construction of a convex outer-approximation of the feasible space by considering $\{(z, p) \in X \times P :$



(a) (b)

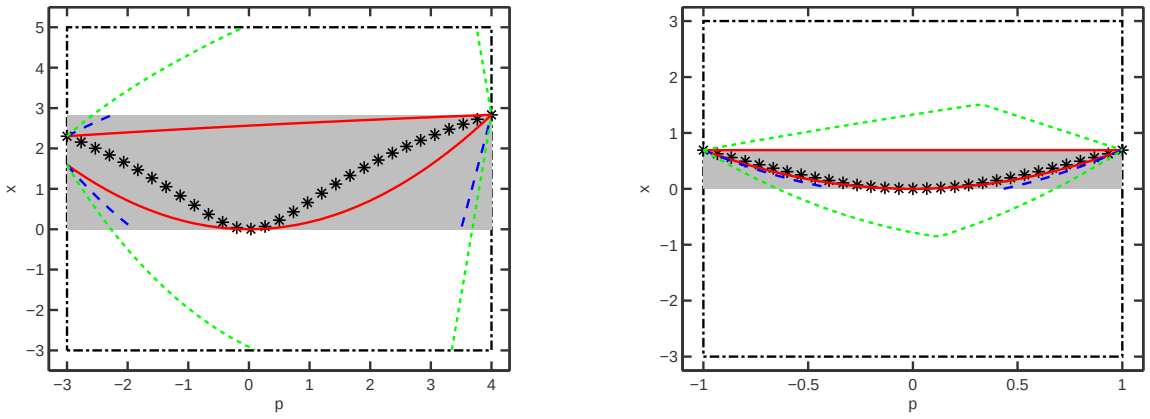
Figure 4.3: Result of reverse McCormick propagation for Example 4.1 showing the original bounds (dashed-dotted lines), the improved bounds (gray box), the convex and concave relaxations (solid red line, respectively) as well as results of the set-valued mapping $x(p)$ (asterisks). In (a) one iteration of the reverse McCormick propagation was performed while in (b) the reverse propagation iterations was repeated ten times. The dashed blue lines show convex and concave relaxations calculated using one iteration of the more expensive method in [164].

$\check{h}(z,p) \leq 0 \leq \hat{h}(z,p)\}$ where $\check{h}, \hat{h}$ are standard McCormick relaxations of h on $X \times P$. This set can be traced by plotting the zero level sets of $\check{h}, \hat{h}$, i.e., $\check{h}(z,p) = 0$ and $\hat{h}(z,p) = 0$. An alternative, tighter outer-approximation can be found by computing $\check{h}, \hat{h}$ on $\tilde{X} \times \tilde{P} = [0, 2.834] \times [-3, 4]$ instead. In Figure 4.4(b), the same information is shown for smaller original intervals, $X = [-3, 3]$ and $P = [-1, 1]$.

Example 4.3. Let $X = [-10, 10]$ and $P = [0, 3]$. Consider $h(z,p) = z^2 - p$ with $(z,p) \in X \times P$. Note that $h(z,p) = 0$ implicitly defines a set-valued mapping $x : P \to \mathbb{P}(X) : p \mapsto \{\sqrt{p}, -\sqrt{p}\}$ so that $h(\xi, p) = 0$ for all $p \in P$ and $\xi \in x(p)$. The results of the reverse McCormick propagation are shown in Figure 4.5. Here, no comparison with [164] is possible due to non-uniqueness of x.

Example 4.4. Let $X = [-10, 10]$ and $P = [0, 3]$. Consider $h(z,p) = z^4 - p^2 + 1$ with $(z,p) \in X \times P$. Note that $h(z,p) = 0$ implicitly defines a set-valued mapping $x : [1,3] \to \mathbb{P}(X) : p \mapsto \{\sqrt[4]{p^2-1}, -\sqrt[4]{p^2-1}\}$ so that $h(\xi, p) = 0$ for all $p \in [1,3]$ and $\xi \in x(p)$. While Figure 4.6(a) shows the result using the original bounds, Figure 4.6(b) depicts the effect of using bounds obtained from reverse interval propagation. In the latter case, the reverse interval propagation reduces both X and P to obtain $\tilde{X}$ and $\tilde{P}$. Then, the reverse McCormick propagation is performed using the reduced intervals $\tilde{X}$ and $\tilde{P}$.

Example 4.5. Let $X = [-10, 10]$ and $P = [-3, 3]$. Consider $h(z,p) = z^2 - (\sqrt{p^2 - p} - 2)^4$ with $(z,p) \in X \times P$. Note that $h(z,p) = 0$ implicitly defines a set-valued mapping



(a) Relaxations for $P = [-3, 4]$ and $X = [-3, 5]$.

(b) Relaxations for $P = [-1, 1]$ and $X = [-3, 3]$.

Figure 4.4: Result of reverse McCormick propagation for Example 4.2 showing the original bounds (dashed-dotted lines), the improved bounds (gray box), the convex and concave relaxations (solid red lines) as well as results of the set-valued mapping $x(p)$ (asterisks). Additionally, zero level sets of the McCormick relaxations of $h(z, p)$ constructed on $X \times P$ (short dashed green lines) as well as $\tilde{X} \times \tilde{P}$ (dashed blue lines) are shown except where they are outside the interval bounds. Here, the results for different $P \times X$ are shown in (a) and (b).

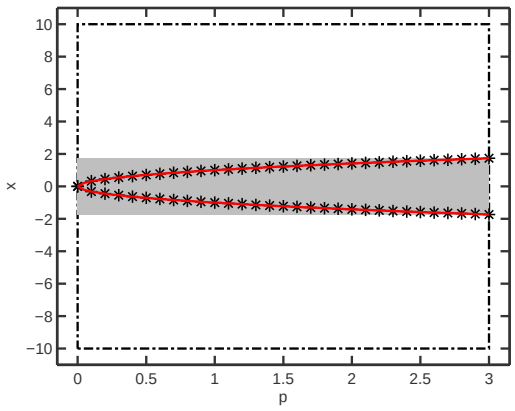


Figure 4.5: Result of reverse McCormick propagation for Example 4.3 showing the original bounds (dashed-dotted lines), the improved bounds (gray box), the convex and concave relaxations (solid red lines) as well as results of the set-valued mapping $x(p)$ (asterisks).

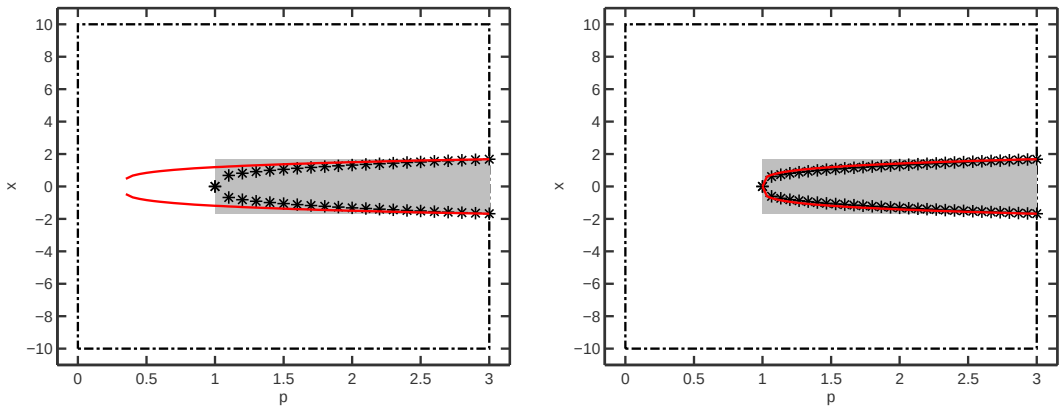


Figure 4.6: Result of reverse McCormick propagation for Example 4.4 showing the original bounds (dashed-dotted lines), the improved bounds (gray box), the convex and concave relaxations (solid red lines) as well as results of the set-valued mapping $x(p)$ (asterisks). While in (a) the original bounds are used, in (b) the result of the bounds obtained from reverse interval propagation is shown.

$x : [-3,0] \cap [1,3] \to \mathbb{P}(X) : p \mapsto \{(\sqrt{p^2-p}-2)^2, -(\sqrt{p^2-p}-2)^2\}$ so that $h(\xi, p) = 0$ for all $p \in [-3,0] \cap [1,3]$ and $\xi \in x(p)$. On the other hand, if $p \in (0,1)$ no feasible z exists that satisfies $h(z,p) = 0$. The results of the reverse McCormick propagation are shown in Figure 4.7. Here, the algorithm was supplied with the information that the argument of the square root cannot be negative.

4.5.2 Inequality constraints

Example 4.6. Let $X = [-10,10]$ and $P = [0,3]$. Consider $h(z,p) = z^2 - p$ and $g(z,p) = (p-1)^2 - z - 2.5$ with $(z,p) \in X \times P$. Note that $h(z,p) = 0$ and $g(z,p) \leq 0$ implicitly defines set-valued mappings $x : [0, 2.03593] \to \mathbb{P}(X) : p \mapsto \{\sqrt{p}, -\sqrt{p}\}$ and $x : (2.03593, 3] \to \mathbb{P}(X) : p \mapsto \{\sqrt{p}\}$ so that $h(\xi, p) = 0$ for all $p \in P$ and $\xi \in x(p)$. However, we are only interested in those (z,p) for which $g(z,p) \leq 0$. The results of the reverse McCormick propagation are shown in Figure 4.8.

4.5.3 Objective function

Example 4.7. Let $Y = [-3,3] \times [-2,2]^2$ and consider the optimization of the six-hump camel back function [52]

$$\begin{aligned} \min_{\mathbf{y} \in Y} \quad & f(\mathbf{y}) = \left(4 - 2.1y_1^2 + \frac{1}{3}y_1^4\right) y_1^2 + y_1 y_2 + \left(-4 + 4y_2^2\right) y_2^2 \\ \text{s.t.} \quad & g(\mathbf{y}) = y_1^2 + (y_2 - 0.5)^2 - 0.5 \leq 0 \end{aligned}$$

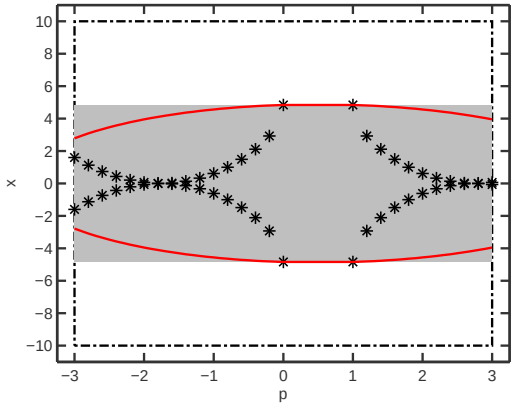


Figure 4.7: Result of reverse McCormick propagation for Example 4.5 showing the original bounds (dotted lines), the improved bounds (dashed lines), the convex and concave relaxations (solid lines) as well as results of the set-valued mapping $x(p)$ (asterisks).

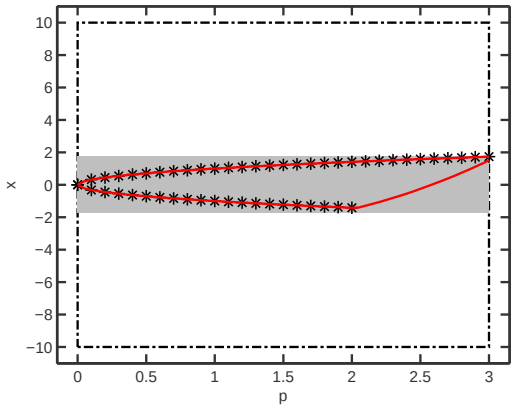
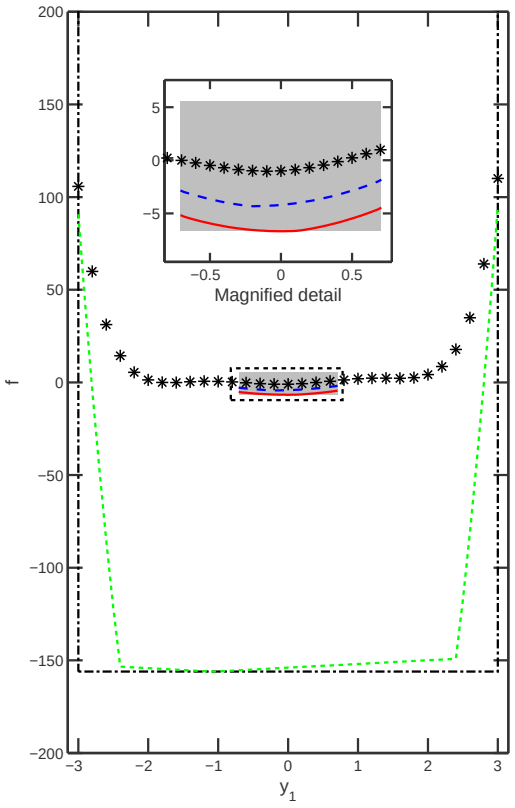


Figure 4.8: Result of reverse McCormick propagation for Example 4.6 showing the original bounds (dotted lines), the improved bounds (gray box), the convex and concave relaxations (solid red lines) as well as results of the set-valued mapping $x(p)$ (asterisks).

where an inequality constraint has been added. We are interested in constructing relaxations of $f(\mathbf{y})$ which take the information from the constraint $g(\mathbf{y}) \leq 0$ into account. Here, let y_1 take the role of the independent and y_2 the role of the dependent variable. The reverse McCormick update will proceed as outlined in Section 4.3.1. Then, one last forward evaluation will be performed to obtain improved relaxations of f. Figure 4.9 shows the obtained relaxations. Clearly, the McCormick relaxations can be improved substantially by incorporating the information from the constraint.

4.6 Conclusion

Reverse McCormick propagation, a new method to construct and improve McCormick relaxations of implicitly defined set-valued mappings has been presented. It takes advantage of the directed acyclic graph representation of a factorable function, which has been previously used for interval calculations [153, 174]. Bounds and relaxations of factors can often be improved by using information about the permissible range of a factorable function and propagating it backwards through the graph. In particular, this allows the construction and improvement of relaxations of mappings that are only implicitly defined. This is useful in the context of CSPs since it allows to construct convex relaxations of non-convex solution sets defined by nonlinear equality constraints and non-convex inequality constraints. Furthermore, McCormick relaxations of the objective function of an NLP can be improved using information contained in the constraints. While Stuber et al. [164] also put forward methods to construct relaxations of implicit functions, the method presented here does not require existence nor uniqueness of the implicit function on all or parts of the domain. Furthermore, it is less computationally intensive and does not require a pre-processing step. It also provides a reduced-space relaxation for nonconvex programs that can take constraints into account, but does not require convex optimizers that can cope with general nonsmooth nonlinear constraints.



(a) Section for $y_2 = 0.7126$ ($y_1 \in [-3, 3]$).

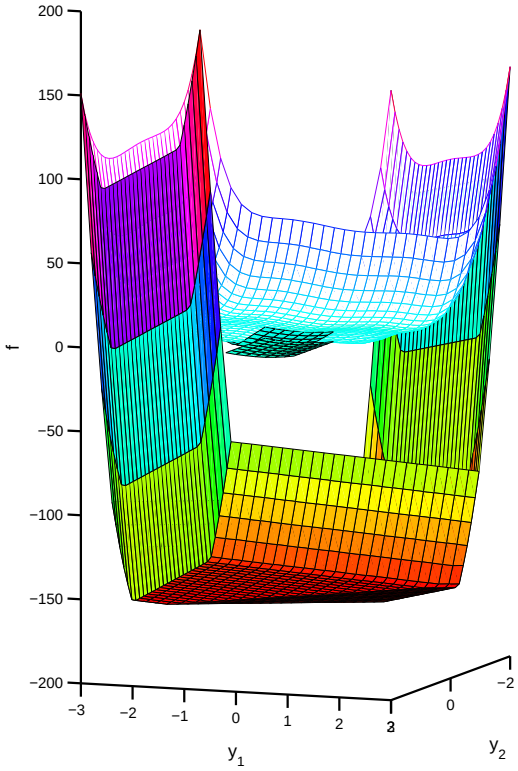


Figure 4.9: Result of reverse McCormick propagation for Example 4.7. In (a) the original bounds (dashed-dotted line), the improved bounds (gray box), the objective function f (asterisks) and the convex relaxations (red line) are shown as well as standard convex McCormick relaxations constructed on Y (green short dashed line) and $\tilde{Y}$ (blue dashed line) in a section. In (b) f is shown as a mesh and relaxations are shown as surfaces.

Chapter 5

Second-order interval bounds for implicit functions

Consider the problem of bounding the parameter dependent zeros of a function on a given domain. In the context of global optimization, we are interested in this problem for two reasons. First, it can be used as a means of domain reduction in the sense of [24, 38, 149]. Second, and more importantly, when considering reduced-space problem formulations [e.g., 56], it provides an initialization for the computation of composite relaxations [156].

Branch-and-bound algorithms for deterministic global optimization [59, 88] require at least quadratically convergent bounds to overcome the cluster problem [54, 178]. While many relaxations for the original space formulation [5, 118, 166] possess this favorable quadratic convergence order [34], this is not necessarily true for the reduced-space approach. When one deviates from the initialization used for standard McCormick relaxations [118], e.g., as indicated by the notion of generalized McCormick relaxations [156], it is possible that this favorable property no longer holds.

The rules for generalized McCormick relaxations require that valid bounds and relaxations are provided for the arguments of a factorable function. Each argument can be thought of as a linear function of all arguments, so that convex and concave relaxations can coincide in this case—as they do in the case of standard McCormick relaxations [118]. Hence, these relaxations possess infinite convergence order in the pointwise sense important for theoretical convergence order considerations [34]. However, if relaxations of implicit functions are to be calculated, e.g., as introduced in Chapter 4 or using the ideas proposed in [162, 164], a different initialization strategy is used: the relaxations of the arguments are initialized as the bounds, which, by definition, are also valid convex and concave relaxations of the argument. As we will see below, many interval techniques possess linear convergence order only, so the results in [34] imply that in this setting the generalized McCormick relaxations will also have linear convergence order only.

If parametric interval methods for systems of equations can be constructed with a higher convergence order, then this information can be used to initialize the generalized McCormick calculations and, thus, to guarantee a higher convergence order of the constructed convex and concave relaxations.

In the past, a variety of interval methods have been proposed for the non-parametric version of bounding the zeros of a function, see [127, Chapter 5] for a review. For the parametric case, Gay [66] proposed a method based on so-called majorizing equations of

the continuous analogue of Newton's method. Neumaier [126] and Rump [147] studied the sensitivity analysis of systems of nonlinear equations with the goal to bound the error rigorously, also see [127, Chapter 5.5]. Recently, Stuber [162, Chapter 3] extended the non-parametric interval methods such as interval Newton and Krawczyk's method to the parametric case.

In this chapter, it is argued that the convergence order of the parametric Krawczyk's method for nonlinear equations is linear in the parameters only. Based on the sensitivities of the nonlinear equations, a second-order convergent method is presented to bound the zeros of a system of nonlinear functions. Also, an initialization strategy is provided that computes initial bounds for the sensitivities, which can then be fed into a linear Gauss-Seidel operator for refinement.

In Section 5.1, important definitions and relevant technical results are collected for reference throughout the remainder of this chapter. In Section 5.2, the convergence of Krawczyk's method for parametric systems of nonlinear equations is studied and in Section 5.3, it is shown how the sensivities can be used to obtain a second-order convergent method. Case studies are reported in Section 5.4 and conclude the chapter.

5.1 Preliminaries

In this chapter, we will consider a $\mathcal{L}$-computational sequence with $n_x + n_p$ inputs and n_x outputs in the sense of Section 3.1. Let $D \subset \mathbb{R}^{n_x+n_p}$ be its natural domain and $\mathbf{f} : D \to \mathbb{R}^{n_x}$ its natural function.

Assumption 5.1. Assume that $D_x \subset \mathbb{R}^{n_x}$ and $D_p \subset \mathbb{R}^{n_p}$ are open and connected, that $D_x \times D_p \subset D$ and that $\mathbf{f}$ is twice continuously differentiable on $D_x \times D_p$.

Let $\mathbf{Jf}(\mathbf{y})$ denote the Jacobian matrix of $\mathbf{f}$ evaluated at $\mathbf{y} \in D$. Often, we will explicitly partition the arguments of $\mathbf{f}$ and write $\mathbf{f}(\mathbf{z}, \mathbf{p})$ where $\mathbf{z} \in D_x$ and $\mathbf{p} \in D_p$. Let $\mathbf{J_x f}(\mathbf{z}, \mathbf{p})$ denote the Jacobian matrix of $\mathbf{f}(\cdot, \mathbf{p})$ evaluated at $\mathbf{z} \in D_x$. Similarly, let $\mathbf{J_p f}(\mathbf{z}, \mathbf{p})$ denote the Jacobian matrix of $\mathbf{f}(\mathbf{z}, \cdot)$ evaluated at $\mathbf{p} \in D_p$.

We are interested in bounding the range of the mapping $\mathbf{x} : P \subset D_p \to \mathbb{R}^{n_x}$, which is defined only implicitly by

$$P \ni \mathbf{p} \mapsto \mathbf{z} \in X : \mathbf{f}(\mathbf{z}, \mathbf{p}) = \mathbf{0}. \tag{5.1}$$

Existence and uniqueness of such a function follows from the result below.

Theorem 5.1. *Let $E \subset \mathbb{R}^{n_x+n_p}$ be an open set and $\mathbf{f} : E \to \mathbb{R}^{n_x}$ be a twice continuously differentiable function so that $\mathbf{f}(\tilde{\mathbf{z}}, \tilde{\mathbf{p}}) = \mathbf{0}$ for some $(\tilde{\mathbf{z}}, \tilde{\mathbf{p}}) \in E$. Assume that $\mathbf{J_x f}(\tilde{\mathbf{z}}, \tilde{\mathbf{p}})$ is invertible. Then, there exist open sets $U \subset \mathbb{R}^{n_x+n_p}$ and $W \subset \mathbb{R}^{n_p}$, $(\tilde{\mathbf{z}}, \tilde{\mathbf{p}}) \in U$ and $\tilde{\mathbf{p}} \in W$, with the following property: to every $\mathbf{p} \in W$ corresponds a unique $\mathbf{z}$ such that $(\mathbf{z}, \mathbf{p}) \in U$ and $\mathbf{f}(\mathbf{z}, \mathbf{p}) = \mathbf{0}$. If this $\mathbf{z}$ is defined to be $\mathbf{x}(\mathbf{p})$ then $\mathbf{x} : W \to \mathbb{R}^{n_x}$ is a twice continuously differentiable function with $\mathbf{x}(\mathbf{p}) = \mathbf{z}$ and $\mathbf{f}(\mathbf{x}(\mathbf{p}), \mathbf{p}) = \mathbf{0}$ for all $\mathbf{p} \in W$.*

Proof. Follows from [81, Theorem 4]. □

5.1.1 Relevant definitions and results from interval analysis

For the purposes of this chapter, we extend Definitions 3.5 and 3.6 for interval vectors to *interval matrices*.

Definition 5.1. We write $A \in \mathbb{IR}^{m\times n}$ to refer to a $m \times n$ matrix whose elements are intervals. Define $\underline{\mathbf{A}}, \overline{\mathbf{A}}, |A|, m(A) \in \mathbb{R}^{m\times m}$ for each $i = 1,\ldots,m$ and $k = 1,\ldots,n$ by $[\underline{\mathbf{A}}]_{ik} = \underline{a}_{ik}$, $[\overline{\mathbf{A}}]_{ik} = \overline{a}_{ik}$, $[|A|]_{ik} = |A_{ik}|$, and $[m(A)]_{ik} = m(A_{ik})$; define $w(A) = \max_{i,k}\{\overline{a}_{ik} - \underline{a}_{ik}\}$. When A is a $n \times n$ interval matrix, we also define $\langle A\rangle \in \mathbb{R}^{n\times n}$ for each $i,k = 1,\ldots,n$ by

$$[\langle A\rangle]_{ik} = \begin{cases} 0 & \text{if } i = k \text{ and } 0 \in A_{ik}, \\ \min\{|\underline{a}_{ik}|, |\overline{a}_{ik}|\} & \text{if } i = k \text{ and } 0 \notin A_{ik}, \\ -|A_{ik}| & \text{otherwise.} \end{cases}$$

For any matrix $A \in \mathbb{IR}^{n\times m}$ or $\mathbf{A} \in \mathbb{R}^{n\times m}$ we use the notation A_j and $\mathbf{A}_j$ to refer to the jth column of A and $\mathbf{A}$, respectively.

We will write $\|\mathbf{A}\| \equiv \|\mathbf{A}\|_\infty$ for $\mathbf{A} \in \mathbb{R}^{n\times n}$.

Definition 5.2. The l_∞ *norms* are defined for $X \in \mathbb{IR}^n$ and $A \in \mathbb{IR}^{n\times n}$ by $\|X\| \equiv \max_i\{|X_i|\}$ and $\|A\| \equiv \max_i\{\sum_k |A_{ik}|\}$. Suppose $\mathbf{u} \in \mathbb{R}^n$ with $\mathbf{u} > \mathbf{0}$. We also define the *scaled maximum norm* for X and A by $\|X\|_{\mathbf{u}} \equiv \max_i\{|X_i|/u_i\}$ and $\|A\|_{\mathbf{u}} \equiv \max_i\{\sum_k |A_{ik}|u_k/u_i\}$.

If $\mathbf{u} = \mathbf{1}$ the scaled maximum norm reduces to the standard l_∞ norm.

Lemma 5.1. *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $X \in \mathbb{IR}^n$. Then, $w(\mathbf{A}X) \le \|\mathbf{A}\|w(X)$.*

Proof. This follows directly from $w(\mathbf{A}X) = \max_i \sum_j w(a_{ij}X_j) = \max_i \sum_j |a_{ij}|w(X_j)$ $\le \max_i \sum_j |a_{ij}|w(X) = \|\mathbf{A}\|w(X)$. □

Lemma 5.2. *[102, p. 191] Let $A \in \mathbb{IR}^{n\times n}$ and $X \in \mathbb{IR}^n$. If $\mathbf{0} \in X$, then $w(AX) \le 2\|A\|w(X)$.*

Lemma 5.3. *Let $\mathbf{I}$ denote the identity matrix in $\mathbb{R}^{n\times n}$, let $A \in \mathbb{IR}^{n\times n}$ and assume that $m(A)$ is invertible. Then $\|\mathbf{I} - m(A)^{-1}A\| \le \frac{1}{2}\|m(A)^{-1}\|nw(A)$.*

Proof. Set $R = \mathbf{I} - m(A)^{-1}A$. Let $\mathbf{Y}^+, \mathbf{Y}^- \in \mathbb{R}^{n\times n}$ be element-wise non-negative matrices so that at most one of corresponding elements is positive and $m(A)^{-1} = \mathbf{Y}^+ - \mathbf{Y}^-$. We can write $\underline{\mathbf{R}} = \mathbf{I} - \mathbf{Y}^+\overline{\mathbf{A}} + \mathbf{Y}^-\underline{\mathbf{A}}$ and $\overline{\mathbf{R}} = \mathbf{I} - \mathbf{Y}^+\underline{\mathbf{A}} + \mathbf{Y}^-\overline{\mathbf{A}}$. It follows that $\overline{\mathbf{R}} = -\underline{\mathbf{R}}$ [cf. 105, Lemma 9] since

$$\begin{aligned} \overline{\mathbf{R}} + \underline{\mathbf{R}} &= \mathbf{I} - \mathbf{Y}^+\underline{\mathbf{A}} + \mathbf{Y}^-\overline{\mathbf{A}} + \mathbf{I} - \mathbf{Y}^+\overline{\mathbf{A}} + \mathbf{Y}^-\underline{\mathbf{A}} \\ &= 2\mathbf{I} - (\mathbf{Y}^+ - \mathbf{Y}^-)\overline{\mathbf{A}} - (\mathbf{Y}^+ - \mathbf{Y}^-)\underline{\mathbf{A}} \\ &= 2\mathbf{I} - m(A)^{-1}(\overline{\mathbf{A}} + \underline{\mathbf{A}}) \\ &= 2\mathbf{I} - m(A)^{-1}2m(A) = \mathbf{0}. \end{aligned}$$

This implies that $\|\overline{\mathbf{R}} - \underline{\mathbf{R}}\| = \|\overline{\mathbf{R}}\| + \|\underline{\mathbf{R}}\| = 2\|\underline{\mathbf{R}}\| = 2\|\overline{\mathbf{R}}\| = 2\|R\|$. Note that $|m(A)^{-1}| = \mathbf{Y}^+ + \mathbf{Y}^-$. Since $\overline{\mathbf{R}} - \underline{\mathbf{R}} = (\mathbf{Y}^+ + \mathbf{Y}^-)(\overline{\mathbf{A}} - \underline{\mathbf{A}}) = |m(A)^{-1}|(\overline{\mathbf{A}} - \underline{\mathbf{A}})$ and $\|\overline{\mathbf{A}} - \underline{\mathbf{A}}\| \le nw(A)$, it follows that $\|\overline{\mathbf{R}} - \underline{\mathbf{R}}\| \le \||m(A)^{-1}|\|\|\overline{\mathbf{A}} - \underline{\mathbf{A}}\| \le \|m(A)^{-1}\|nw(A)$ and the result follows. □

Lemma 5.4. *Let $A \in \mathbb{IR}^{m\times n}$. If $\mathbf{0} \in A$ then $\|A\| \leq nw(A)$.*

Proof. $\|A\| = \max_i \sum_k |A_{ik}| \leq \max_i \sum_k w(A_{ik}) \leq w(A) \max_i \sum_k 1 = nw(A)$. □

Definition 5.3. Let $A \in \mathbb{IR}^{m\times n}$. A is *regular* if every real matrix $\mathbf{A} \in A$ has rank n. Suppose that A is a square matrix and that $m(A)$ is regular then A is called *strongly regular* if $m(A)^{-1}A$ is regular.

We will always tacitly assume that A strongly regular implies $m(A)$ regular.

Definition 5.4. Let $A \in \mathbb{IR}^{n\times n}$. If $A_{ik} \leq 0$ for all $i \neq k$ and there exists some positive $\mathbf{u} \in \mathbb{R}^n$ such that $A\mathbf{u} > \mathbf{0}$, then we call A an *M-matrix*. If $\langle A \rangle$ is an M-matrix, then A is called an *H-matrix*.

See Section 3.6 and 3.7 in [127] for results that characterize M- and H-matrices. Here, we shall only point out that any H-matrix is regular [127, 3.7.5].

Definition 5.5. The *linear interval equation* with coefficient matrix $A \in \mathbb{IR}^{m\times n}$ and right-hand side $B \in \mathbb{IR}^m$ is defined as the family of linear equations $\mathbf{Ax} = \mathbf{b}$ for any $\mathbf{A} \in A$ and $\mathbf{b} \in B$. Its *solution set*[1] is given by $\Sigma(A, B) \equiv \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{b} \text{ for some } \mathbf{A} \in A \text{ and } \mathbf{b} \in B\}$. If A is regular, define the *hull of the solution set* $A^{\mathrm{H}}B = \mathrm{hull}\{\Sigma(A, B)\}$ for any $B \in \mathbb{IR}^n$. If A is additionally a $n \times n$ interval matrix, then define the *hull inverse* $A^{\mathrm{H}} : \mathbb{IR}^n \to \mathbb{IR}^n$ by $A^{\mathrm{H}}B = \mathrm{hull}\{\mathbf{A}^{-1}\mathbf{b} : \mathbf{A} \in A, \mathbf{b} \in B\}$ for any $B \in \mathbb{IR}^n$. The notation extends directly to the case when B is a conformable interval matrix.

Definition 5.6. Let $\Gamma : \mathbb{IR} \times \mathbb{IR} \times \mathbb{IR} \to \mathbb{IR}$ for all $A, B, X \in \mathbb{IR}$ by $\Gamma(A, B, X) = \mathrm{hull}\{x \in X : \exists a \in A, b \in B : ax = b\}$. Define the *Gauss-Seidel operator* $\Gamma : \mathbb{IR}^{n\times n} \times \mathbb{IR}^n \times \mathbb{IR}^n \to \mathbb{IR}^n$ for all $A \in \mathbb{IR}^{n\times n}$ and $B, X \in \mathbb{IR}^n$ by $\Gamma(A, B, X) = \mathbf{y}$ where

$$y_i = \Gamma\left(A_{ii}, B_i - \sum_{k=1}^{i-1} A_{ik}y_k - \sum_{k=i+1}^{n} A_{ik}X_k, X_i\right),\ i = 1, \ldots, k.$$

For a characterization of $\Gamma : \mathbb{IR} \times \mathbb{IR} \times \mathbb{IR} \to \mathbb{IR}$ see Proposition 4.1.

Let $J_{\mathbf{x}}\mathbf{f} : \mathfrak{D}_x \times \mathfrak{D}_p \to \mathbb{IR}^{n_x\times n_x}$ and $J_{\mathbf{p}}\mathbf{f} : \mathfrak{D}_x \times \mathfrak{D}_p \to \mathbb{IR}^{n_x\times n_p}$ denote inclusion functions of $\mathbf{J}_{\mathbf{x}}\mathbf{f}$ and $\mathbf{J}_{\mathbf{p}}\mathbf{f}$ on $\mathfrak{D}_x \times \mathfrak{D}_p$, respectively.

Lemma 5.5. *If $Z \subset \mathbb{R}^n$ is compact then $\mathbb{I}Z$ is compact.*

Proof. Note that $\mathbb{H}^{2n} = \{(\mathbf{a}, \mathbf{b}) \in \mathbb{R}^n \times \mathbb{R}^n : \mathbf{a} \leq \mathbf{b}\}$ and $\mathbb{IR}^n$ are a metric spaces [155, p. 82]. Also, $\pi : \mathbb{H}^{2n} \ni (\mathbf{a}, \mathbf{b}) \mapsto [\mathbf{a}, \mathbf{b}] \in \mathbb{IR}^n$ is continuous since it is an isometry [155, cf. Lemma 2.5.2]. Let $Z^{\mathbb{H}} = \{(\mathbf{a}, \mathbf{b}) \in \mathbb{H}^{2n} : [\mathbf{a}, \mathbf{b}] \in Z\}$, which is compact. Note that $\pi(Z^{\mathbb{H}}) = \mathbb{I}Z$. Since the image of a compact metric space under a continuous function is compact [145, Theorem 4.14], the result follows. □

[1] Note that the solution set of a linear interval equation is usually not an interval [127, p. 92] and often not even a convex set.

Lemma 5.6. *Let $X \in \mathbb{IR}^n$ and consider $F : \mathbb{I}X \to \mathbb{IR}^{m\times m}$. If F is locally Lipschitz on $\mathbb{I}X$ and $m(F(\tilde{X}))$ is invertible for any $\tilde{X} \in \mathbb{I}X$ then there exists $\mathbf{M}' \in \mathbb{R}^{m\times m}$ so that $|[m(F(\tilde{X}))]^{-1}| \leq \mathbf{M}'$ for any $\tilde{X} \in \mathbb{I}X$.*

Proof. $m(F(\cdot))$ is continuous on $\mathbb{I}X$ since F is locally Lipschitz on $\mathbb{I}X$, which implies continuity, $m(\cdot)$ is continuous and the composition of continuous functions is continuous. By assumption, $[m(F(\tilde{X}))]^{-1}$ exists for any $\tilde{X} \in \mathbb{I}X$. Furthermore, the inverse operator and the magnitude operator are continuous so that $|[m(F(\cdot))]^{-1}|$ is continuous. Since X is compact, $\mathbb{I}X$ is compact and $|[m(F(\cdot))]^{-1}|$ is bounded on $\mathbb{I}X$. □

Definition 5.7. Suppose $\mathbf{A} \in \mathbb{R}^{n\times n}$. The maximal magnitude of the eigenvalues of $\mathbf{A}$ denoted by $\rho(\mathbf{A})$ is called the *spectral radius* of $\mathbf{A}$.

The reader is also reminded of results in Section 3.2.2 where centered forms and similar inclusion functions are discussed. There, it is established that the centered form is a second-order convergent inclusion function.

5.2 Convergence order of parametric interval Newton methods

Interval analysis provides several methods for the task of bounding the solution set of a system of nonlinear equations. In their typical form, they are designed to enclose zeros of functions or provide a guarantee that no zero exists on the considered interval [127, cf. Chapter 5]. Parametric interval Newton methods are discussed in [162, Chapter 3]. As an exemplar, we will look closer at the convergence order of Krawczyk's methods; other methods include the Hansen-Sengupta operator [73].

Definition 5.8. Let $X \in \mathbb{I}D_x$, $P \in \mathbb{I}D_p$, $\mathbf{x} \in X$ and $\mathbf{Y}(X,P) \in \mathbb{R}^{n_x\times n_x}$. The *parametric Krawczyk operator* $K : \mathbb{R}^{n_x} \times \mathbb{IR}^{n_x} \times \mathbb{IR}^{n_p} \to \mathbb{IR}^{n_x}$ is defined as

$$K(\mathbf{x}, X, P) = X \cap (\mathbf{x} - \mathbf{Y}(X,P)F([\mathbf{x},\mathbf{x}],P) + [\mathbf{I} - \mathbf{Y}(X,P)J_{\mathbf{x}}\mathbf{f}(X,P)]\,(X - \mathbf{x}))\,.$$

It is well known that the *non-parametric* Krawczyk operator (i.e., when $P = [\mathbf{p},\mathbf{p}]$) has quadratic convergence order.

Theorem 5.2. *[cf. 123] Let $X \in \mathbb{I}D_x$. Assume that $\mathbf{J}_{\mathbf{x}}\mathbf{f}$ is non-singular in a neighborhood of the solution, in particular $m(J_{\mathbf{x}}\mathbf{f}(\tilde{X}))$ is non-singular for any $\tilde{X} \in \mathbb{I}X$, that $J_{\mathbf{x}}\mathbf{f}$ is locally Lipschitz on $\mathbb{I}X$ and that $\mathbf{Y}(\tilde{X}) = \left[m(J_{\mathbf{x}}\mathbf{f}(\tilde{X}))\right]^{-1} + \mathbf{E}(\tilde{X})$ where $\|\mathbf{E}(\tilde{X})\| \leq Cw(\tilde{X})$ with $C \geq 0$ for any $\tilde{X} \in \mathbb{I}X$. Then, the convergence of the non-parametric Krawczyk operator is quadratic, i.e., there exists some $q > 0$ so that*

$$w(K(\mathbf{z},\tilde{X})) \leq qw(\tilde{X})^2,\ \forall \tilde{X} \in \mathbb{I}X, \mathbf{z} \in \tilde{X}.$$

Proof. Let $\tilde{X} \in \mathbb{I}X$ and $\mathbf{z} \in \tilde{X}$. Define $R \equiv \mathbf{I} - \mathbf{Y}(\tilde{X})J_{\mathbf{x}}\mathbf{f}(\tilde{X})$ and note that $w(K(\mathbf{z},\tilde{X})) \leq w(R(\tilde{X} - \mathbf{z})) \leq 2\|R\|w(\tilde{X} - \mathbf{z}) = 2\|R\|w(\tilde{X})$ where the second inequality follows from

Lemma 5.2. Let L be the Lipschitz constant of $J_{\mathbf{x}}\mathbf{f}$ so that $w(J_{\mathbf{x}}\mathbf{f}(\tilde{X})) \leq Lw(\tilde{X})$, cf. Theorem 3.2. For shorter notation, set $\mathbf{M} = \left[m(J_{\mathbf{x}}\mathbf{f}(\tilde{X}))\right]^{-1}$. Now, we have $\|R\| \leq \|\mathbf{I} - \mathbf{M}J_{\mathbf{x}}\mathbf{f}(\tilde{X})\| + \|\mathbf{E}(\tilde{X})J_{\mathbf{x}}\mathbf{f}(\tilde{X})\| \leq \frac{1}{2}\|\mathbf{M}\|n_x w(J_{\mathbf{x}}\mathbf{f}(\tilde{X})) + \|\mathbf{E}(\tilde{X})\|\|J_{\mathbf{x}}\mathbf{f}(\tilde{X})\| \leq \frac{1}{2}\|\mathbf{M}\|n_x Lw(\tilde{X}) + Cw(\tilde{X})\|J_{\mathbf{x}}\mathbf{f}(X)\|$ where the second inequality follows from Lemma 5.3. As $|\mathbf{M}|$ is bounded on $\mathbb{I}X$ by some $\mathbf{M}'$, see Lemma 5.6, $q = n_x L\|\mathbf{M}'\| + 2C\|J_{\mathbf{x}}\mathbf{f}(X)\|$ is independent of $\mathbf{z}, \tilde{X}$ and it follows that $w(K(\mathbf{z}, \tilde{X})) \leq qw(\tilde{X})^2$ for any $\tilde{X} \in \mathbb{I}X$ and $\mathbf{z} \in \tilde{X}$.

□

The result, and especially its proof, indicate that it will be impossible to establish a quadratic convergence rate result for the parametric case in an analogue fashion. More precisely, when $P \in \mathbb{I}D_p$ is a non-degenerate interval, it is obvious that the interval hull of $\mathbf{x}(P) \equiv \{\mathbf{x} \in D_x : \exists \mathbf{p} \in P, \mathbf{f}(\mathbf{x}, \mathbf{p}) = \mathbf{0}\}$ is, in general, a non-degenerate interval as well. Consequently, given a non-degenerate P, convergence to a degenerate interval is impossible. This observation is formalized in the following theorem.

Theorem 5.3. *Let $X \in \mathbb{I}D_x$ and $P \in \mathbb{I}D_p$. Assume that $\mathbf{J}_{\mathbf{x}}\mathbf{f}$ is non-singular in a neighborhood of the solution, in particular $m(J_{\mathbf{x}}\mathbf{f}(\tilde{X}, \tilde{P}))$ is non-singular for any $\tilde{X} \in \mathbb{I}X$, $\tilde{P} \in \mathbb{I}P$. Further, suppose that $F([\mathbf{x}, \mathbf{x}], \cdot)$ is locally Lipschitz on $\mathbb{I}P$ for each $\mathbf{x} \in X$, that $J_{\mathbf{x}}\mathbf{f}$ is locally Lipschitz on $\mathbb{I}X \times \mathbb{I}P$ and that $\mathbf{Y}(\tilde{X}, \tilde{P}) = \left[m(J_{\mathbf{x}}\mathbf{f}(\tilde{X}, \tilde{P}))\right]^{-1} + \mathbf{E}(\tilde{X}, \tilde{P})$ where $\|\mathbf{E}(\tilde{X}, \tilde{P})\| \leq C_x w(\tilde{X}) + C_p w(\tilde{P})$ with $C_x, C_p \geq 0$ for any $\tilde{X} \in \mathbb{I}X$, $\tilde{P} \in \mathbb{I}P$. Then, the convergence of the Krawczyk operator is quadratic in X and linear in P, i.e., there exists some $q_1, q_2 > 0$ so that*

$$w(K(\mathbf{z}, \tilde{X}, \tilde{P})) \leq q_1 w(\tilde{X})^2 + q_2 w(\tilde{P}),\ \forall \tilde{X} \in \mathbb{I}X, \tilde{P} \in \mathbb{I}P, \mathbf{z} \in \tilde{X}.$$

Proof. Let $\tilde{X} \in \mathbb{I}X$, $\mathbf{z} \in X$ and $\tilde{P} \in \mathbb{I}P$. For easier notation define $R \equiv \mathbf{I} - \mathbf{Y}(\tilde{X}, \tilde{P})J_{\mathbf{x}}\mathbf{f}(\tilde{X}, \tilde{P})$ and note that $w(K(\mathbf{z}, \tilde{X}, \tilde{P})) \leq w(\mathbf{Y}(\tilde{X}, \tilde{P})F([\mathbf{z}, \mathbf{z}], \tilde{P})) + w(R(\tilde{X} - \mathbf{z}))$. Lemma 5.1 yields that $w(\mathbf{Y}(\tilde{X}, \tilde{P})F([\mathbf{z}, \mathbf{z}], \tilde{P})) \leq \|\mathbf{Y}(\tilde{X}, \tilde{P})\|w(F([\mathbf{z}, \mathbf{z}], \tilde{P}))$. Lemma 5.2 implies that $w(R(\tilde{X} - \mathbf{z})) \leq 2\|R\|w(\tilde{X} - \mathbf{z}) = 2\|R\|w(\tilde{X})$. Let $L_{J_{\mathbf{x},1}}$ and $L_{J_{\mathbf{x},2}}$ be the Lipschitz constants of $J_{\mathbf{x}}\mathbf{f}$ on $\mathbb{I}X \times \mathbb{I}P$ so that $w(J_{\mathbf{x}}\mathbf{f}(\tilde{X}, \tilde{P})) \leq L_{J_{\mathbf{x},1}} w(\tilde{X}) + L_{J_{\mathbf{x},2}} w(\tilde{P})$ and $L_{f_{\mathbf{x}}}$ and $L_{f_{\mathbf{p}}}$ be the Lipschitz constants of F on $\mathbb{I}X \times \mathbb{I}P$ so that $w(F(\tilde{X}, \tilde{P})) \leq L_{f_{\mathbf{x}}} w(\tilde{X}) + L_{f_{\mathbf{p}}} w(\tilde{P})$, cf. Theorem 3.2. In particular, $w(F([\mathbf{z}, \mathbf{z}], \tilde{P})) \leq L_{f_{\mathbf{p}}} w(\tilde{P})$. For shorter notation, set $\mathbf{M} = \left[m(J_{\mathbf{x}}\mathbf{f}(\tilde{X}, \tilde{P}))\right]^{-1}$. Now, we have $\|\mathbf{Y}(\tilde{X}, \tilde{P})\|w(F([\mathbf{z}, \mathbf{z}], \tilde{P})) \leq \|\mathbf{M}\|L_{f_{\mathbf{p}}} w(\tilde{P}) + (C_x w(\tilde{X}) + C_p w(\tilde{P}))L_{f_{\mathbf{p}}} w(\tilde{P}))$ and $\|R\| \leq \|\mathbf{I} - \mathbf{M}J_{\mathbf{x}}\mathbf{f}(\tilde{X}, \tilde{P})\| + \|\mathbf{E}(\tilde{X}, \tilde{P})J_{\mathbf{x}}\mathbf{f}(\tilde{X}, \tilde{P})\| \leq \frac{1}{2}\|\mathbf{M}\|n_x w(J_{\mathbf{x}}\mathbf{f}(\tilde{X}, \tilde{P})) + \|\mathbf{E}(\tilde{X}, \tilde{P})\|\|J_{\mathbf{x}}\mathbf{f}(\tilde{X}, \tilde{P})\| \leq \frac{1}{2}\|\mathbf{M}\|n_x(L_{J_{\mathbf{x},1}} w(\tilde{X}) + L_{J_{\mathbf{x},2}} w(\tilde{P})) + (C_x w(\tilde{X}) + C_p w(\tilde{P}))\|J_{\mathbf{x}}\mathbf{f}(X, P)\|$ where the second inequality follows from Lemma 5.3. As $|\mathbf{M}|$ is bounded on $\mathbb{I}X \times \mathbb{I}P$ by some $\mathbf{M}'$, see Lemma 5.6, $q_1 = \|\mathbf{M}'\|n_x L_{J_{\mathbf{x},1}} + 2C_x\|J_{\mathbf{x}}\mathbf{f}(X, P)\|$ and $q_2 = \|\mathbf{M}'\|(L_{f_{\mathbf{p}}} + n_x L_{J_{\mathbf{x},2}} w(X)) + L_{f_{\mathbf{p}}}(C_x w(X) + C_p w(P)) + 2C_p\|J_{\mathbf{x}}\mathbf{f}(X, P)\|w(X)$ are independent of $\mathbf{z}, \tilde{X}, \tilde{P}$ and it follows that $w(K(\mathbf{z}, \tilde{X}, \tilde{P})) \leq q_1 w(\tilde{X})^2 + q_2 w(\tilde{P})$ for any $\tilde{X} \in \mathbb{I}X$, $\tilde{P} \in \mathbb{I}P$ and $\mathbf{z} \in \tilde{X}$. □

Next, we provide an example where the parametric Krawczyk method (and the parametric Hansen-Sengupta operator) achieves linear convergence order only.

Example 5.1. [cf. 162, Example 4.5.3] Let $n_x = 5$ and $n_p = 3$. Let $D_x \times D_p = \mathbb{R}^5 \times \mathbb{R}^3$ and

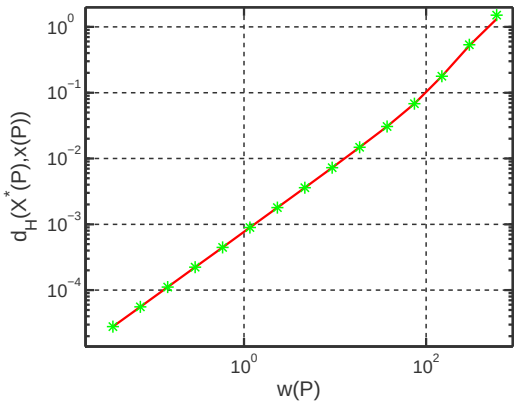


Figure 5.1: Empirical convergence order of the parametric Hansen-Sengupta operator (red line) and the parametric Krawczyk operator (green asterisks) for Example 5.1

consider

$$\mathbf{f}(\mathbf{z},\mathbf{p}) = \begin{bmatrix} 0 - z_1 + \Delta t \left(k_1 z_4 z_5 - c_{O_2}(p_1+p_2)z_1 + \frac{p_1}{K_2} z_3 + \frac{p_2}{K_3} z_2 - k_5 z_1^2 \right) \\ 0 - z_2 + \Delta t \left(p_2 c_{O_2} z_1 - \left(\frac{p_2}{K_3} + p_3 \right) z_2 \right) \\ 0 - z_3 + \Delta t \left(p_1 c_{O_2} z_1 - \frac{p_1}{K_2} z_3 \right) \\ 0.4 - z_4 + \Delta t \left(-k_{1s} z_4 z_5 \right) \\ 140 - z_5 + \Delta t \left(-k_1 z_4 z_5 \right) \end{bmatrix}$$

where $\Delta t = 0.01$, $T = 273$, $K_2 = 46 \exp(\frac{6500}{T} - 18)$, $K_3 = 2K_2$, $k_1 = 53$, $k_{1s} = 10^{-6} k_1$, $k_5 = 0.0012$ and $c_{O_2} = 0.02$. Let $X = [0,140]^3 \times [0,0.4] \times [0,140]$ and $P = [10,1200]^2 \times [0.001,40]$. Let $P^1 = P$ and $P_i^l = m(P_i^{l-1}) + \frac{1}{4}[-w(P_i^{l-1}), w(P_i^{l-1})]$, $i = 1,\ldots,3$ and $l = 2,\ldots,15$. For each P^l, the image of P^l under $\mathbf{x}$ was numerically estimated using BARON [150] in GAMS by minimizing or maximizing z_i while requiring $(\mathbf{z},\mathbf{p}) \in X \times P^l$ and $\mathbf{f}(\mathbf{z},\mathbf{p}) = \mathbf{0}$ for $i = 1,\ldots,5$. $X^*(P^l)$ denotes the approximate limit of the parametric Krawczyk operator on P^l, which is calculated by stopping the iteration when $d_H(X^k, K(m(X^k), X^k, P^l)) < 10^{-8}$. Each inclusion function was evaluated using its natural interval extension. $\mathbf{Y}(X,P^1) = [m(J_{\mathbf{x}}(X,P^1))]^{-1}$ and $\mathbf{Y}(X,P^l) = [m(J_{\mathbf{x}}(X^*(P^{l-1}),P^l))]^{-1}$, $l > 1$ were used as preconditioners.

In Figure 5.1, $d_H(X^*(P^l), \mathbf{x}(P^l))$ is plotted against $w(P^l)$ showing that the convergence order of the parametric Krawczyk operator (and the parametric Hansen-Sengupta operator) is linear only.

We conclude this section with a convergence result for the parametric Krawczyk method based on a theorem given in [162].

Theorem 5.4. *Assume that we use the natural interval extensions as inclusion functions. Suppose that* $\{P^l\} \subset \mathbb{ID}_p$ *defines a nested sequence of intervals such that* $\bigcap_{l=1}^{\infty} P^l = [\mathbf{p}^*, \mathbf{p}^*]$. *Let* $X \in \mathbb{ID}_x$ *be such that there exists a unique solution* $\mathbf{x}(\mathbf{p}) \in X$ *for every* $\mathbf{p} \in P^1$. *Suppose that* $m(J_{\mathbf{x}}\mathbf{f}(\tilde{X},\tilde{P}))$

is non-singular for any $\tilde{X} \in \mathbb{I}X$ and $\tilde{P} \in P^1$. Let $\{X^k(P^l)\} \subset \mathbb{I}X$ be the nested sequence of intervals given by $X^1(P^l) = K(m(X), X, P^l)$ and $X^{k+1}(P^l) = K(m(X^k(P^l)), X^k(P^l), P^l)$ with $\mathbf{Y}(\tilde{X}, \tilde{P}) = [m(J_{\mathbf{x}}\mathbf{f}(\tilde{X}, \tilde{P}))]^{-1}$. Let $\lambda^* = \|[m(J_{\mathbf{x}}\mathbf{f}(X, P^1)]^{-1} J_{\mathbf{x}}\mathbf{f}(X, P^1) - \mathbf{I}\|$. If $\lambda^* < \frac{1}{2}$ then

$$\lim_{k\to\infty}\lim_{l\to\infty} X^k(P^l) = \lim_{l\to\infty}\lim_{k\to\infty} X^k(P^l) = [\mathbf{x}(\mathbf{p}^*), \mathbf{x}(\mathbf{p}^*)].$$

Proof. Since $\lambda^* < \frac{1}{2}$, $J_{\mathbf{x}}\mathbf{f}(X, P^1)$ is strongly regular [127, 4.1.1], which implies that $J_{\mathbf{x}}\mathbf{f}(\tilde{X}, P^l)$ is strongly regular for any l and $\tilde{X} \in \mathbb{I}X$ [127, 4.1.3].

Next, we will argue that $K : D_x \times \mathbb{I}D_x \times \mathbb{I}D_x \to \mathbb{I}\mathbb{R}^{n_x}$ is continuous on its domain. Since $\mathbf{f}$ is twice continuously differentiable on $D_x \times D_p$, $\mathbf{J}_{\mathbf{x}}\mathbf{f}$ is continuously differentiable. As a result, $\mathbf{f}$ and $\mathbf{J}_{\mathbf{x}}\mathbf{f}$ are locally Lipschitz on $D_x \times D_p$ and, in particular, locally Lipschitz on any compact subset of their respective domains. This implies that F and $J_{\mathbf{x}}\mathbf{f}$ are locally Lipschitz on any compact subset of their respective domains [127, 2.1.1]. Furthermore, $[m(J_{\mathbf{x}}\mathbf{f}(\tilde{X}, \tilde{P}))]^{-1}$ is continuous on $X \times P^1$ as shown in the proof of Lemma 5.6. Thus, K is locally Lipschitz on its domain [127, 2.1.1], and thus also continuous.

Continuity of K on its domain implies that $X^k(\cdot)$ is continuous on $\mathbb{I}P^1$. Consequently, it follows that $\lim_{k\to\infty}\lim_{l\to\infty} X^k(P^l) = \lim_{k\to\infty} X^k([\mathbf{p}^*, \mathbf{p}^*])$ so that it suffices to consider convergence of the non-parametric case. $\lambda^* < \frac{1}{2}$ implies that $\rho(|[m(J_{\mathbf{x}}\mathbf{f}(X, P^1)]^{-1} J_{\mathbf{x}}\mathbf{f}(X, P^1) - \mathbf{I}|) < \frac{1}{2}$ [127, 3.2.3]. From strong regularity of $J_{\mathbf{x}}\mathbf{f}(X^k([\mathbf{p}^*, \mathbf{p}^*]), [\mathbf{p}^*, \mathbf{p}^*])$ we conclude that $\lim_{k\to\infty} X^k([\mathbf{p}^*, \mathbf{p}^*]) = [\mathbf{x}(\mathbf{p}^*), \mathbf{x}(\mathbf{p}^*)]$ [127, 5.2.2].

Let $L_{f_{\mathbf{p}}}$ be such that $w(F(m(\tilde{X}), \tilde{P})) \leq L_{f_{\mathbf{p}}} w(\tilde{P})$ for all $\tilde{P} \in \mathbb{I}P$ and $\tilde{X} \in \mathbb{I}X$, which is guaranteed to exist as F is Lipschitz. By Lemma 5.6, there exists some $\mathbf{M}' \in \mathbb{R}^{n_x \times n_x}$ such that $\|\mathbf{Y}(X^k(P^l), P^l)\| \leq \|\mathbf{M}'\|$. Set $R(X^k(P^l), P^l) = \mathbf{I} - \mathbf{Y}(X^k(P^l), P^l) J_{\mathbf{x}}\mathbf{f}(X^k(P^l), P^l)$ to shorten notation. Note that $m(R(X^k(P^l), P^l)) = \mathbf{0}$. Thus, $w(R(X^k(P^l), P^l)(X^k(P^l) - m(X^k(P^l)))) \leq 2\|R(X^k(P^l), P^l)\| w(X^k(P^l) - m(X^k(P^l)))$. We have

$$\begin{aligned}
w(X^{k+1}(P^l)) &\leq w(\mathbf{Y}(X^k(P^l), P^l) F(m(X^k(P^l)), P^l) + R(X^k(P^l), P^l)(X^k(P^l) - m(X^k(P^l)))) \\
&\leq w(\mathbf{Y}(X^k(P^l), P^l) F(m(X^k(P^l)), P^l)) \\
&\quad \ldots + w(R(X^k(P^l), P^l)(X^k(P^l) - m(X^k(P^l)))) \\
&\leq \|\mathbf{Y}(X^k(P^l), P^l)\| w(F(m(X^k(P^l)), P^l)) \\
&\quad \ldots + 2\|R(X^k(P^l), P^l)\| w(X^k(P^l) - m(X^k(P^l))) \\
&\leq \|\mathbf{M}'\| L_{f_{\mathbf{p}}} w(P^l) + 2\lambda^* w(X^k(P^l)).
\end{aligned}$$

Let $k \to \infty$ to find that $w(X^*(P^l)) = \lim_{k\to\infty} w(X^k(P^l)) \leq \|\mathbf{M}'\| L_{f_{\mathbf{p}}} w(P^l) + 2\lambda^* w(X^*(P^l))$. Since $0 \leq \lambda^* < \frac{1}{2}$, $w(X^*(P^l)) \leq \frac{\|\mathbf{M}'\| L_{f_{\mathbf{p}}}}{1-2\lambda^*} w(P^l)$. Thus, $\{X^*(P^l)\}$ converges to a degenerate interval as $l \to \infty$. Since $\mathbf{x}(\mathbf{p}^*) \in X^k(P^l)$, it follows that $\lim_{l\to\infty} X^*(P^l) = [\mathbf{x}^*(\mathbf{p}^*), \mathbf{x}^*(\mathbf{p}^*)]$. □

5.3 Sensitivity-based bounding method

Theorem 3.5 indicates that centered forms are interval methods to bound the range of a function with quadratic convergence order. Since we are interested in bounding the range of $\mathbf{x}$ as defined in Equation (5.1) with quadratic convergence order, centered forms suggest one such a method. For mean value forms, which are a particular kind of centered form, it is necessary to bound the range of the Jacobian of $\mathbf{x}$. Unfortunately, $\mathbf{x}$ is only defined implicitly, so usually no closed, factorable expression for it exists from which a locally Lipschitz inclusion function of $\frac{\partial x_i}{\partial p_j}$ could be obtained by combining automatic differentiation [70] with the rules of interval arithmetic. The following result demonstrates a bound on $\mathbf{x}$ can be obtained instead.

Proposition 5.1. *[cf. 126] Consider $\tilde{P} \in \mathbb{I}P$ and $\tilde{X} \in \mathbb{I}D_x$. Let $\tilde{\mathbf{p}} \in \tilde{P}$ and suppose there exists $\tilde{\mathbf{z}} \in \tilde{X}$ so that $\mathbf{f}(\tilde{\mathbf{z}}, \tilde{\mathbf{p}}) = \mathbf{0}$. Assume that $J_{\mathbf{x}}\mathbf{f}(\tilde{X}, [\tilde{\mathbf{p}}, \tilde{\mathbf{p}}])$ is a regular interval matrix and suppose $S \in \mathbb{IR}^{n_x \times n_p}$ encloses $-\Sigma(J_{\mathbf{x}}\mathbf{f}(\tilde{X}, [\tilde{\mathbf{p}}, \tilde{\mathbf{p}}]), J_{\mathbf{p}}\mathbf{f}(\tilde{X}, \tilde{P}))$. Then,*

$$\{\mathbf{z} \in \tilde{X} : \exists \mathbf{p} \in \tilde{P}, \mathbf{f}(\mathbf{z}, \mathbf{p}) = \mathbf{0}\} \subset \tilde{\mathbf{z}} + S(\tilde{P} - \tilde{\mathbf{p}}).$$

Proof. Let $\mathbf{z} \in \tilde{X}$ such that there exists $\mathbf{p} \in \tilde{P}$ so that $\mathbf{f}(\mathbf{z}, \mathbf{p}) = \mathbf{0}$. For each $i = 1, \ldots, n_x$, the mean value theorem guarantees that there exists $\lambda_i, \mu_i \in [0, 1]$ so that

$$\begin{aligned} f_i(\mathbf{z}, \mathbf{p}) &= f_i(\mathbf{z}, \tilde{\mathbf{p}}) + \sum_{k=1}^{n_p} \frac{\partial f_i}{\partial p_k}(\mathbf{z}, \mathbf{p} + \mu_i(\tilde{\mathbf{p}} - \mathbf{p}))(p_k - \tilde{p}_k) \\ &= f_i(\tilde{\mathbf{z}}, \tilde{\mathbf{p}}) + \sum_{j=1}^{n_x} \frac{\partial f_i}{\partial x_j}(\mathbf{z} + \lambda_i(\tilde{\mathbf{z}} - \mathbf{z}), \tilde{\mathbf{p}})(z_j - \tilde{z}_j) + \sum_{k=1}^{n_p} \frac{\partial f_i}{\partial p_k}(\mathbf{z}, \mathbf{p} + \mu_i(\tilde{\mathbf{p}} - \mathbf{p}))(p_k - \tilde{p}_k) \\ 0 &= \sum_{j=1}^{n_x} \frac{\partial f_i}{\partial x_j}(\mathbf{z} + \lambda_i(\tilde{\mathbf{z}} - \mathbf{z}), \tilde{\mathbf{p}})(z_j - \tilde{z}_j) + \sum_{k=1}^{n_p} \frac{\partial f_i}{\partial p_k}(\mathbf{z}, \mathbf{p} + \mu_i(\tilde{\mathbf{p}} - \mathbf{p}))(p_k - \tilde{p}_k). \end{aligned}$$

Let $\mathbf{J}_{\mathbf{x},i}\mathbf{f}$ and $\mathbf{J}_{\mathbf{p},i}\mathbf{f}$ refer to the ith row of $\mathbf{J}_{\mathbf{x}}\mathbf{f}$ and $\mathbf{J}_{\mathbf{p}}\mathbf{f}$, respectively, so that we can write

$$\mathbf{J}_{\mathbf{x},i}\mathbf{f}(\mathbf{z} + \lambda_i(\tilde{\mathbf{z}} - \mathbf{z}), \tilde{\mathbf{p}})(\mathbf{z} - \tilde{\mathbf{z}}) + \mathbf{J}_{\mathbf{p},i}\mathbf{f}(\mathbf{z}, \mathbf{p} + \mu_i(\tilde{\mathbf{p}} - \mathbf{p}))(\mathbf{p} - \tilde{\mathbf{p}}) = 0.$$

Next, introduce $\tilde{\mathbf{J}}_{\mathbf{x}} \in \mathbb{R}^{n_x \times n_x}$ and $\tilde{\mathbf{J}}_{\mathbf{p}} \in \mathbb{R}^{n_x \times n_p}$ where each row $i = 1, \ldots, n_x$ is defined by $\tilde{\mathbf{J}}_{\mathbf{x},i} = \mathbf{J}_{\mathbf{x},i}\mathbf{f}(\mathbf{z} + \lambda_i(\tilde{\mathbf{z}} - \mathbf{z}), \tilde{\mathbf{p}})$ and $\tilde{\mathbf{J}}_{\mathbf{p},i} = \mathbf{J}_{\mathbf{p},i}\mathbf{f}(\mathbf{z}, \mathbf{p} + \mu_i(\tilde{\mathbf{p}} - \mathbf{p}))$. Since $\mathbf{z} + \lambda_i(\tilde{\mathbf{z}} - \mathbf{z}) \in \tilde{X}$ for each i, $\tilde{\mathbf{J}}_{\mathbf{x}} \in J_{\mathbf{x}}\mathbf{f}(\tilde{X}, [\tilde{\mathbf{p}}, \tilde{\mathbf{p}}])$ so that $\tilde{\mathbf{J}}_{\mathbf{x}}$ is invertible and we have

$$\mathbf{z} = \tilde{\mathbf{z}} - \tilde{\mathbf{J}}_{\mathbf{x}}^{-1}\tilde{\mathbf{J}}_{\mathbf{p}}(\mathbf{p} - \tilde{\mathbf{p}}).$$

Note that $\tilde{\mathbf{J}}_{\mathbf{x}}^{-1}\tilde{\mathbf{J}}_{\mathbf{p}} \in \Sigma(J_{\mathbf{x}}\mathbf{f}(\tilde{X}, [\tilde{\mathbf{p}}, \tilde{\mathbf{p}}]), J_{\mathbf{p}}\mathbf{f}(\tilde{X}, \tilde{P}))$ and the result follows. □

Remark 5.1. Interestingly, it is only necessary to take the interval extension of $\mathbf{J}_{\mathbf{x}}\mathbf{f}$ with respect to $\mathbf{z}$, but not to $\mathbf{p}$, cf. the Hansen-Sengupta operator [73, 127]. Also, by changing the order in which the mean value theorem is applied, one can alternatively obtain that S

should enclose $-\Sigma(J_{\mathbf{x}}\mathbf{f}(\tilde{X},\tilde{P}), J_{\mathbf{p}}\mathbf{f}([\tilde{\mathbf{z}},\tilde{\mathbf{z}}],\tilde{P}))$. The disadvantage of this formulation is that $J_{\mathbf{x}}\mathbf{f}(\tilde{X},\tilde{P}) \supset J_{\mathbf{x}}\mathbf{f}(\tilde{X},[\tilde{\mathbf{p}},\tilde{\mathbf{p}}])$.

Next, we will consider the convergence order of the proposed inclusion function of $\mathbf{x}$.

Proposition 5.2. *Let $P \in \mathbb{IR}_p$. Suppose that $S : \mathbb{IR}\times P \to \mathbb{IR}^{n_x\times n_p}$ is given so that, for each $\tilde{P} \in \mathbb{IR}$ and $\tilde{\mathbf{p}} \in \tilde{P}$, $S(\tilde{P},\tilde{\mathbf{p}}) \supset -\Sigma(J_{\mathbf{x}}\mathbf{f}(\mathrm{hull}(\mathbf{x}(\tilde{P})),[\tilde{\mathbf{p}},\tilde{\mathbf{p}}]), J_{\mathbf{p}}\mathbf{f}(\mathrm{hull}(\mathbf{x}(\tilde{P})),\tilde{P}))$. Assume that there exists $L > 0$ so that $w(S(\tilde{P},\tilde{\mathbf{p}})) \leq Lw(\tilde{P})$ for all $\tilde{P} \in \mathbb{IR}$ and $\tilde{\mathbf{p}} \in \tilde{P}$. Then, there exists some $q > 0$ so that*

$$d_H(x_i(\tilde{P})), x_i(\tilde{\mathbf{p}}) + S_i(\tilde{P},\tilde{\mathbf{p}})(\tilde{P}-\tilde{\mathbf{p}})) \leq qw(\tilde{P})^2, \quad \forall i \in 1,\ldots,n_x, \tilde{P} \in \mathbb{IR}, \tilde{\mathbf{p}} \in \tilde{P}.$$

Proof. Since $\max_i |\tilde{P}_i - \tilde{p}_i| \leq w(\tilde{P})$ for any $\tilde{\mathbf{p}} \in \tilde{P}$, the result follows from Theorem 3.5. □

Assumption 5.2. Assume that for each $\mathbf{p} \in D_p$ there exists exactly one $\mathbf{z} \in D_x$ so that $\mathbf{f}(\mathbf{z},\mathbf{p}) = \mathbf{0}$.

For the remainder of the paper, we restrict our attention to $P \in \mathbb{ID}_p$. Furthermore, assume that $\mathbf{x}(P) \subset \check{X}$ for some $\check{X} \in \mathbb{ID}_x$. Since the inclusion functions in Proposition 5.1 also depend on $\tilde{X} \in \mathbb{ID}_x$, it proves convenient to consider a crude inclusion function of $\mathbf{x}$ first. Let $X : \mathbb{IR} \to \mathbb{IR}^{n_x}$ be defined for any $\tilde{P} \in \mathbb{IR}$ by $X(\tilde{P}) = \mathbf{x}(m(\tilde{P})) + \check{S}(\tilde{P} - m(\tilde{P}))$ where $\check{S} \in \mathbb{IR}^{n_x\times n_p}$ so that $-\Sigma(J_{\mathbf{x}}\mathbf{f}(\check{X},P), J_{\mathbf{p}}\mathbf{f}(\check{X},P)) \subset \check{S}$. Then, Proposition 5.1 implies that $\mathbf{x}(\tilde{P}) \subset X(\tilde{P})$ for any $\tilde{P} \in \mathbb{IR}$. Furthermore, it is immediately clear that there exists $L > 0$ so that $w(X(\tilde{P})) \leq Lw(\tilde{P})$ for all $\tilde{P} \in \mathbb{IR}$. As an aside: the important property of X is that its width is linear in $w(\tilde{P})$, in principle, any bounding method on $\mathbf{x}$ that satisfies this property can be used.

Given Propositions 5.1 and 5.2, it remains to discuss how we can obtain $S : \mathbb{IR} \to \mathbb{IR}^{n_x\times n_p}$ that bounds $-\Sigma(J_{\mathbf{x}}\mathbf{f}(X(\tilde{P}),m(\tilde{P})), J_{\mathbf{p}}\mathbf{f}(X(\tilde{P}),\tilde{P}))$ and has the following property: there exists $L > 0$ so that $w(S(\tilde{P})) \leq Lw(\tilde{P})$ for all $\tilde{P} \in \mathbb{IR}$. The definition of X presents an opportunity to simplify our notation hereafter as we can drop the explicit dependence of any inclusion function on $\check{X}$ and consider its dependence on $\tilde{P}$ only. To simplify notation hereafter, we will discuss the following results in terms of interval functions $A : \mathbb{IR} \to \mathbb{IR}^{n_x\times n_x}$ and $B : \mathbb{IR} \to \mathbb{IR}^{n_x\times n_p}$ instead, which can be thought of as placeholders for $J_{\mathbf{x}}\mathbf{f}(X(\cdot),m(\cdot))$ and $-J_{\mathbf{p}}\mathbf{f}(X(\cdot),\cdot)$, respectively.

5.3.1 Obtaining an initial enclosure of the sensitivities

Before Krawczyk's method for linear systems [102] or the interval Gauss-Seidel operator [142] can be used to refine the enclosure of $\Sigma(A(\tilde{P}),B_j(\tilde{P}))$, $j = 1,\ldots,n_p$, for a given $\tilde{P} \in \mathbb{IR}$, it is necessary to obtain an initial valid bound for this enclosure. Theorem 4.4.10 in [127] suggests a good initialization strategy when $A(\tilde{P})$ is strongly regular. Taking this result in consideration, Neumaier proposed an algorithm, which has been adapted here as Algorithm 5.1, for obtaining an initial enclosure of the solution set of a linear interval equation. In our case, it requires solving n_p linear systems with n_x equations each. The obtained S^0 can be improved, e.g., by using the preconditioned Gauss-Seidel operator as discussed in the following section.

Lemma 5.7. *Assume that $A(\tilde{P})$ is strongly regular for any $\tilde{P} \in \mathbb{P}$. Let $\tilde{\mathbf{Z}} : \mathbb{P} \to \mathbb{R}^{n_x \times n_p}$ be given so that for each $\tilde{P} \in \mathbb{P}$ there exists some $\tilde{\mathbf{A}}(\tilde{P}) \in A(\tilde{P})$ and $\tilde{\mathbf{B}}(\tilde{P}) \in B(\tilde{P})$ with $\tilde{\mathbf{A}}(\tilde{P})\tilde{\mathbf{Z}}(\tilde{P}) = \tilde{\mathbf{B}}(\tilde{P})$. Suppose that A and B are locally Lipschitz on $\mathbb{P}$. Let $\mathbf{Y}(\tilde{P}) = m(A(\tilde{P}))^{-1}$. Fix $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{n_x}$ so that $\mathbf{u}, \mathbf{v} > \mathbf{0}$ and $\langle \mathbf{Y}(\tilde{P})A(\tilde{P}) \rangle \mathbf{u} \geq \mathbf{v}$ for all $\tilde{P} \in \mathbb{P}$. Set $S_j^0(\tilde{P}) = \tilde{\mathbf{Z}}_j(\tilde{P}) + \|\mathbf{Y}(\tilde{P})(B_j(\tilde{P}) - A(\tilde{P})\tilde{\mathbf{Z}}_j(\tilde{P}))\|_{\mathbf{v}}[-\mathbf{u}, \mathbf{u}]$ for any $\tilde{P} \in \mathbb{P}$ and $j = 1, \ldots, n_p$. Then, $S^0(\tilde{P}) \supset (A(\tilde{P}))^{\mathrm{H}} B(\tilde{P})$ for any $\tilde{P} \in \mathbb{P}$. Also, there exists some $L_S > 0$ so that $w(S_j^0(\tilde{P})) \leq L_S w(\tilde{P})$ for all $\tilde{P} \in \mathbb{P}$ and $j = 1, \ldots, n_p$.*

Proof. Since $A(\tilde{P})$ is strongly regular, $\mathbf{Y}(\tilde{P})A(\tilde{P})$ is an H-matrix [127, 4.1.1]. By definition, $\langle \mathbf{Y}(\tilde{P})P(\tilde{A}) \rangle$ is an M-matrix and by [127, 3.7.3] regular. Furthermore, the referred to $\mathbf{u}$ and $\mathbf{v}$ exist [127, 3.7.1 and p. 112]. Since $\mathbf{Y}(\tilde{P})A(\tilde{P})$ is regular for any $\tilde{P} \in \mathbb{P}$, we have

$$\begin{aligned} A(\tilde{P})^{\mathrm{H}} B_j(\tilde{P}) &\subset (\mathbf{Y}(\tilde{P})A(\tilde{P}))^{\mathrm{H}} (\mathbf{Y}(\tilde{P})B_j(\tilde{P})) \\ &\subset \tilde{\mathbf{Z}}_j(\tilde{P}) + (\mathbf{Y}(\tilde{P})A(\tilde{P}))^{\mathrm{H}} (\mathbf{Y}(\tilde{P})B_j(\tilde{P}) - \mathbf{Y}(\tilde{P})A(\tilde{P})\tilde{\mathbf{Z}}_j(\tilde{P})) \\ &= \tilde{\mathbf{Z}}_j(\tilde{P}) + (\mathbf{Y}(\tilde{P})A(\tilde{P}))^{\mathrm{H}} (\mathbf{Y}(\tilde{P})(B_j(\tilde{P}) - A(\tilde{P})\tilde{\mathbf{Z}}_j(\tilde{P}))) \\ &\subset \tilde{\mathbf{Z}}_j(\tilde{P}) + \|\mathbf{Y}(\tilde{P})(B_j(\tilde{P}) - A(\tilde{P})\tilde{\mathbf{Z}}_j(\tilde{P}))\|_{\mathbf{v}}[-\mathbf{u}, \mathbf{u}] = S_j^0(\tilde{P}) \end{aligned}$$

for any $j = 1, \ldots, n_p$, where the first inclusion follows from [127, 4.1.5], the second from [127, 4.2.1], the third line follows from [127, 3.1.2] since $\mathbf{Y}(\tilde{P}) \in \mathbb{R}^{n_x \times n_x}$ and the third inclusion from [127, 4.1.9]. Let $\tilde{P} \in \mathbb{P}$ and pick $j \in \{1, \ldots, n_p\}$. Note that $B_j(\tilde{P}) - A(\tilde{P})\tilde{\mathbf{Z}}_j(\tilde{P}) = \left[B_j(\tilde{P}) - \tilde{\mathbf{B}}_j(\tilde{P})\right] - \left[A(\tilde{P})\tilde{\mathbf{Z}}_j(\tilde{P}) - \tilde{\mathbf{B}}_j(\tilde{P})\right]$ and that $\mathbf{0} \in \left[B_j(\tilde{P}) - \tilde{\mathbf{B}}_j(\tilde{P})\right]$, $\mathbf{0} \in \left[A(\tilde{P})\tilde{\mathbf{Z}}_j(\tilde{P}) - \tilde{\mathbf{B}}_j(\tilde{P})\right]$, thus $\mathbf{0} \in \mathbf{Y}(\tilde{P})(B_j(\tilde{P}) - A(\tilde{P})\tilde{\mathbf{Z}}_j(\tilde{P}))$ as $\mathbf{Y}(\tilde{P})$ is regular. Note that $\tilde{\mathbf{Z}}_j(\tilde{P})$ is bounded since $A(\tilde{P})$ is regular and hence $\Sigma(A(\tilde{P}), B(\tilde{P})) \ni \tilde{\mathbf{Z}}_j(\tilde{P})$ is bounded [127, p. 93]. Also, $\|\mathbf{Y}(\tilde{P})\|$ is bounded by Lemma 5.6. Since there exist positive L_A and L_{B_j} so that $w(A(\tilde{P})) \leq L_A w(\tilde{P})$ and $w(B_j(\tilde{P})) \leq L_{B_j} w(\tilde{P})$ and since $\|\tilde{\mathbf{Z}}_j(\tilde{P})\|$ is bounded, it follows with Lemmas 5.4 and 5.1 that there exists some $L_j > 0$ so that $\|\mathbf{Y}(\tilde{P})(B_j(\tilde{P}) - A(\tilde{P})\tilde{\mathbf{Z}}_j(\tilde{P}))\| \leq L_j w(\tilde{P})$. □

The result is a formalization of the argument in [127, p. 124f] and provides the theoretical foundation for Algorithm 5.1. It is important to point out that if $\mathbf{u}, \mathbf{v}$ satisfy $\langle \mathbf{Y}(\tilde{P})A(\tilde{P}) \rangle \mathbf{u} \geq \mathbf{v}$ then they will also satisfy $\langle \mathbf{Y}(P')A(P') \rangle \mathbf{u} \geq \mathbf{v}$ for any $P' \subset \mathbb{I}\tilde{P}$. For shorter notation, set $B'(\tilde{P}) = \mathbf{Y}(\tilde{P})(B_j(\tilde{P}) - A(\tilde{P})\tilde{\mathbf{Z}}_j(\tilde{P}))$. For any $\varepsilon > 0$, $\mathbf{v} = |B'(\tilde{P})| + \varepsilon\mathbf{1} > \mathbf{0}$. Thus, we can construct $\mathbf{u} > \mathbf{0}$ by solving the linear system $\langle \mathbf{Y}(\tilde{P})A(\tilde{P}) \rangle \mathbf{u} = \mathbf{v}$. It remains to compute $\|B'(\tilde{P})\|_{\mathbf{v}}$. However, note that $\|B'(\tilde{P})\|_{\mathbf{v}} \leq \frac{1}{\alpha}$ is equivalent to $|B'(\tilde{P})| \leq \frac{1}{\alpha}\mathbf{v}$ [127, p. 85]. Since $\mathbf{v} = \langle \mathbf{Y}(\tilde{P})A(\tilde{P}) \rangle \mathbf{u}$, an estimate of $\|B'(\tilde{P})\|_{\mathbf{v}}$ can be obtained by finding the smallest $\alpha > 0$ that satisfies $\langle \mathbf{Y}(\tilde{P})A(\tilde{P}) \rangle \mathbf{u} \geq \alpha |B'(\tilde{P})|$. Note that $\alpha \approx 1$ if $\varepsilon\mathbf{1} \ll |B'(\tilde{P})|$.

Lastly, as pointed out in [127, p. 124], this procedure provides a crude enclosure only. In the next section, more advanced interval methods will be applied to the task of refining this enclosure.

Algorithm 5.1: Obtaining an initial enclosure of the sensitivities [cf. 127, p. 150]

Input: $\tilde{X} \in \mathbb{ID}_x, \tilde{P} \in \mathbb{IP}, \varepsilon > 0$
Output: Enclosure of the solution set $S \in \mathbb{IR}^{n_x \times n_p}$ **or** A is not strongly regular

$A \leftarrow J_{\mathbf{x}}\mathbf{f}(\tilde{X}, \tilde{P})$, $B \leftarrow -J_{\mathbf{p}}\mathbf{f}(\tilde{X}, \tilde{P})$;
$\mathbf{Y} \leftarrow [m(A)]^{-1}$;
$\tilde{\mathbf{Z}} \leftarrow \mathbf{Y}m(B)$;
$A' \leftarrow \mathbf{Y}A$, $B' \leftarrow \mathbf{Y}(B - A\tilde{\mathbf{Z}})$;
for $j = 1, \ldots, n_p$ **do**
  Find $\mathbf{u} \in \mathbb{R}^{n_x}$, $\mathbf{u} > \mathbf{0}$ so that $\langle A' \rangle \mathbf{u} = [|B'_{1j}| + \varepsilon \ldots |B'_{n_x j}| + \varepsilon]^{\mathrm{T}}$;
  if $\mathbf{u}$ *does not exist* **then**
    return (A is not strongly regular);
  Find smallest $\alpha > 0$ so that $\langle A' \rangle \mathbf{u} \geq \alpha |B'_j|$;
  $S_j \leftarrow \tilde{\mathbf{Z}}_j + \frac{1}{\alpha}[-\mathbf{u}, \mathbf{u}]$
return (S);

5.3.2 Improving the sensitivity bound

We will use Algorithm 5.1 only to establish an initial bound $S^0(P)$. For any $\tilde{P} \in \mathbb{IP}$, we can set $S^0(\tilde{P}) = S(\tilde{P}^2)$ where $S(\tilde{P}^2)$ has been calculated for some $\tilde{P}^2 \in \mathbb{IP}$, $\tilde{P}^2 \supset \tilde{P}$. Preconditioned interval Gauss-Seidel [142] or Krawczyk's method [102] can be used directly to further improve $S^0(\tilde{P})$. We will show that this provides a linearly convergent bound on $-\Sigma(A(\tilde{P}), B_j(\tilde{P}))$. Since the iterates of the preconditioned interval Gauss-Seidel are guaranteed to be tighter enclosures than those obtained from Krawczyk's method [127, 4.3.5], showing that the latter yields a linearly convergent parametric bound suffices.

Lemma 5.8. *Suppose that $m(A(\tilde{P}))$ is regular for any $\tilde{P} \in \mathbb{IP}$ and there exists some $L > 0$ such that $w(A(\tilde{P})) \leq Lw(\tilde{P})$ for all $\tilde{P} \in \mathbb{IP}$. Then, there exists $L' > 0$ so that $\|I - m(A(\tilde{P}))^{-1}A(\tilde{P})\| \leq L'n_x w(\tilde{P})$ for all $\tilde{P} \in \mathbb{IP}$.*

Proof. First, note that $m(A(\tilde{P}))^{-1}$ exists for all $\tilde{P} \in \mathbb{IP}$. The midpoint operation, the inverse of a regular real matrix and interval matrix multiplication are locally Lipschitz functions so that there exists some $L' > 0$ such that $w(m(A(\tilde{P}))^{-1}A(\tilde{P})) \leq L'w(\tilde{P})$ for all $\tilde{P} \in \mathbb{IP}$. Pick any $\tilde{P} \in \mathbb{IP}$. Note that $\mathbf{0} \in \mathbf{I} - m(A(\tilde{P}))^{-1}A(\tilde{P})$ since $\mathbf{I} \in m(A(\tilde{P}))^{-1}A(\tilde{P})$. Now, Lemma 5.4 provides that $\|\mathbf{I} - m(A(\tilde{P}))^{-1}A(\tilde{P})\| \leq n_x w(\mathbf{I} - m(A(\tilde{P}))^{-1}A(\tilde{P})) = n_x w(m(A(\tilde{P}))^{-1}A(\tilde{P}))$ from which the assertion follows. □

Lemma 5.9. *Suppose that $A(\tilde{P})$ is strongly regular for any $\tilde{P} \in \mathbb{IP}$ and that there exists some $L > 0$ such that $w(A(\tilde{P})) \leq Lw(\tilde{P})$ and $w(B(\tilde{P})) \leq Lw(\tilde{P})$ for all $\tilde{P} \in \mathbb{IP}$. Let $\mathbf{Y}(\tilde{P}) = [m(A(\tilde{P}))]^{-1}$. Then, there exists some $L' > 0$ such that $w((\mathbf{Y}(\tilde{P})A(\tilde{P}))^H(\mathbf{Y}(\tilde{P})B(\tilde{P}))) \leq L'w(\tilde{P})$ for all $\tilde{P} \in \mathbb{IP}$.*

Proof. Note that the hull inverse is a sublinear mapping [127, 3.5.1] and each sublinear mapping is locally Lipschitz as a consequence of [127, 3.5.3]. Then, apply Theorem 3.2 to obtain the result. □

We will use the Krawczyk iteration for linear interval equations [127, Sec. 4.2] to improve $S^0(\tilde{P})$ for any $\tilde{P} \in \mathbb{I}P$. Let $S_j^{k+1}(\tilde{P}) = S_j^k(\tilde{P}) \cap (\mathbf{Y}(\tilde{P})B_j(\tilde{P}) - (\mathbf{Y}(\tilde{P})A(\tilde{P}) - \mathbf{I})S_j^k(\tilde{P}))$ for any $k > 1$ and $j = 1, \ldots, n_p$ where $\mathbf{Y}(\tilde{P}) = [m(A(\tilde{P}))]^{-1}$. Denote the limit of this iteration as $K^*(\tilde{P})$.

Proposition 5.3. *Suppose that $A(\tilde{P})$ is strongly regular for any $\tilde{P} \in \mathbb{I}P$ and that there exists some $L > 0$ such that $w(A(\tilde{P})) \leq Lw(\tilde{P})$ and $w(B(\tilde{P})) \leq Lw(\tilde{P})$ for all $\tilde{P} \in \mathbb{I}P$. Assume that $\|\mathbf{I} - m(A(\tilde{P}))^{-1}A(\tilde{P})\| = \beta(\tilde{P}) < 1$ for all $\tilde{P} \in \mathbb{I}P$. Then, there exists some $L' > 0$ such that $w(K^*(\tilde{P})) \leq L'w(\tilde{P})$ for all $\tilde{P} \in \mathbb{I}P$.*

Proof. Pick any $\tilde{P} \in \mathbb{I}P$. Note that Theorem 4.2.4 in [127] implies that $w(K^*(\tilde{P})) \leq \frac{1+\beta(\tilde{P})}{1-\beta(\tilde{P})} w(A(\tilde{P})^{\mathrm{H}}B(\tilde{P}))$. Together with Lemma 5.9, it follows that $w(K^*(\tilde{P})) \leq \frac{1+\beta(\tilde{P})}{1-\beta(\tilde{P})} L''w(\tilde{P})$ for some $L'' > 0$. However, since $\beta(\tilde{P})$ is bounded for all $\tilde{P} \in \mathbb{I}P$ by assumption, there exists some $L' > 0$ so that we have $w(K^*(\tilde{P})) \leq L'w(\tilde{P})$. □

Note that Lemma 5.8 indicates that for each nested sequence $\{P^l\} \subset \mathbb{I}P$ that converges to a degenerate interval, there exists some $l_1 > 0$ so that $\beta(P^l) < 1$ holds true for all $l > l_1$.

5.3.3 Second-order convergent bounding method

Lemma 5.7 already presented a first linearly convergent method, but the obtained enclosures can be rather crude [127, p. 124]. In a practical implementation Algorithm 5.1 will be executed only once for P and for any subsequent $\tilde{P} \in \mathbb{I}P$ an iterative procedure to improve $S^0(\tilde{P})$ will be used.

Note that Proposition 5.3 is a result about the limit of the Krawczyk iteration. Any implementation, however, will terminate finitely. Thus, it is also necessary to guarantee $w(S^{k+1}(\tilde{P})) \leq L'w(\tilde{P})$ in this case, which results from the following proposition.

Proposition 5.4. *Suppose that $A(\tilde{P})$ is strongly regular for any $\tilde{P} \in \mathbb{I}P$ and that there exists some $L > 0$ such that $w(A(\tilde{P})) \leq Lw(\tilde{P})$ and $w(B(\tilde{P})) \leq Lw(\tilde{P})$ for all $\tilde{P} \in \mathbb{I}P$. Let $\mathbf{Y}(\tilde{P}) = [m(A(\tilde{P}))]^{-1}$. Suppose $S^k(\tilde{P})$ denotes the kth iterate of the linear Krawczyk iteration and assume that $S^0(\tilde{P}) \supset A(\tilde{P})^H B(\tilde{P})$ is bounded for all $\tilde{P} \in \mathbb{I}P$. Then, there exists $L' > 0$ so that $w(S^{k+1}(\tilde{P})) \leq L'w(\tilde{P})$ for all $k > 0$.*

Proof. Pick any $k > 0$ and $j \in \{1, \ldots, n_p\}$. $S_j^{k+1}(\tilde{P}) \subset A(\tilde{P})^{\mathrm{H}}B_j(\tilde{P}) + (\mathbf{Y}(\tilde{P})A(\tilde{P}) - \mathbf{I})(S_j^k(\tilde{P}) - S_j^k(\tilde{P}))$ holds [127, 4.2.3]. Consequently, $w(S^{k+1}(\tilde{P})) \leq w(A(\tilde{P})^{\mathrm{H}}B_j(\tilde{P})) + w((\mathbf{Y}(\tilde{P})A(\tilde{P}) - \mathbf{I})(S_j^k(\tilde{P}) - S_j^k(\tilde{P})))$. Lemma 5.9 implies that there exists $L' > 0$ so that $w(A(\tilde{P})^{\mathrm{H}}B_j(\tilde{P})) \leq L'w(\tilde{P})$ and $w((\mathbf{Y}(\tilde{P})A(\tilde{P}) - \mathbf{I})(S_j^k(\tilde{P}) - S_j^k(\tilde{P}))) \leq 2\|\mathbf{Y}(\tilde{P})A(\tilde{P}) - \mathbf{I}\| w(S_j^k(\tilde{P}) - S_j^k(\tilde{P}))$ by Lemma 5.2. Now, $w(S_j^k(\tilde{P}) - S_j^k(\tilde{P})) \leq w(S_j^0(\tilde{P}) - S_j^0(\tilde{P}))$, which is bounded by assumption, say by $c_j > 0$. Lastly, Lemma 5.8 implies that there exists $L'' > 0$ so that $2\|\mathbf{Y}(\tilde{P})A(\tilde{P}) - \mathbf{I}\| \leq L''w(\tilde{P})$. Hence, $w(S_j^{k+1}(\tilde{P})) \leq (L' + L''c_j)w(\tilde{P})$. □

A first possible implementation of the proposed method is given as Algorithm 5.2. In the beginning of Section 5.3, we pointed out the need for an inclusion function of $\mathbf{x}$ with

$w(X(\tilde{P}) \leq Lw(\tilde{P})$. Instead of the crude inclusion function given there, we will instead rely on the method established thereafter.

By replacing the Krawczyk operator with the Gauss-Seidel operator for linear interval equations, see Definition 5.6, which is known to yield tighter bounds than Krawczyk's method [127, 4.3.5] and by updating prior to each iteration of the Gauss-Seidel operator A and B, it is possible to further improve the resulting bounds at some additional computational expense. This improved method is given as Algorithm 5.3.

The result below summarizes the important second-order convergence result of the either method.

Theorem 5.5. *Let $P \in \mathbb{ID}_p$ and $X \in \mathbb{ID}_x$ so that $J_{\mathbf{x}}\mathbf{f}(X, P)$ is strongly regular. Furthermore, let $J_{\mathbf{x}}\mathbf{f}$ and $J_{\mathbf{p}}\mathbf{f}$ be locally Lipschitz on $\mathbb{IP} \times \mathbb{IX}$. Let $S^0(\tilde{P})$ be given as in Lemma 5.7 and suppose that some $\tilde{S}^0(\tilde{P}) \subset S^0$, $\tilde{S}^0(\tilde{P}) \supset -\Sigma(J_{\mathbf{x}}\mathbf{f}(\mathrm{hull}(\mathbf{x}(\tilde{P})), m(\tilde{P})), J_{\mathbf{p}}\mathbf{f}(\mathrm{hull}(\mathbf{x}(\tilde{P})), \tilde{P}))$ is used to initialize Krawczyk's method for linear interval equations. The iteration is terminated finitely. Let $K^{\dagger}(\tilde{P})$ denote the result of the finite Krawczyk iteration for each $\tilde{P} \in \mathbb{IP}$. Then, there exists some $q > 0$ so that*

$$d_H(x_i(\tilde{P}), x_i(m(\tilde{P})) + K_i^{\dagger}(\tilde{P})(\tilde{P} - m(\tilde{P}))) \leq qw(\tilde{P})^2, \quad \forall i = 1, \ldots, n_x, \tilde{P} \in \mathbb{IP}.$$

Proof. Since $J_{\mathbf{x}}\mathbf{f}(X, P)$ is strongly regular and $J_{\mathbf{x}}\mathbf{f}$, $J_{\mathbf{p}}\mathbf{f}$ are locally Lipschitz on $\mathbb{IP} \times \mathbb{IX}$, the assumptions of Proposition 5.4 are satisfied. Thus, the assumptions of Proposition 5.2 hold and the result follows. □

Algorithm 5.2: Second-order convergent bounding method for parametric nonlinear equations

Input: $\tilde{X} \in \mathbb{ID}_x, \tilde{P} \in \mathbb{IP}, \varepsilon > 0, S \in \mathbb{IR}^{n_x \times n_p}$[optional]
Output: Refined X and S **or** A is not strongly regular

if *No S provided* **then**
  $S \leftarrow$ InitialBound$(\tilde{X}, \tilde{P}, \varepsilon)$; `// uses Algorithm 5.1`
  if InitialBound *failed* **then**
    return (A is not strongly regular);
Find $\mathbf{x} \in \tilde{X}$ so that $\mathbf{f}(\mathbf{x}, m(\tilde{P})) = \mathbf{0}$;
$A \leftarrow J_{\mathbf{x}}\mathbf{f}(\tilde{X}, m(\tilde{P}))$, $B \leftarrow -J_{\mathbf{p}}\mathbf{f}(\tilde{X}, \tilde{P})$;
$A \leftarrow \mathbf{Y}A$, $B \leftarrow \mathbf{Y}B$;
repeat
  $\tilde{X}^{\mathrm{old}} = \tilde{X}$;
  for $i = 1, \ldots, n_p$ **do**
    $S_i \leftarrow S_i \cap (B_i - (A - \mathbf{I})S_i)$;
  $\tilde{X} \leftarrow \tilde{X} \cap \mathbf{x} + S(\tilde{P} - m(\tilde{P}))$;
until $d_H(\tilde{X}, \tilde{X}^{old}) < \varepsilon$;
return $(\tilde{X}, S)$;

Algorithm 5.3: Improved second-order convergent bounding method for parametric nonlinear equations

Input: $\tilde{X} \in \mathbb{ID}_x, \tilde{P} \in \mathbb{IP}, \varepsilon > 0, S \in \mathbb{IR}^{n_x \times n_p}$[optional]
Output: Refined X and S **or** A is not strongly regular

if *No S provided* **then**
  $S \leftarrow$ InitialBound$(\tilde{X}, \tilde{P}, \varepsilon)$; `// uses Algorithm 5.1`
  if InitialBound *failed* **then**
    return (A is not strongly regular);
Find $\mathbf{x} \in \tilde{X}$ so that $\mathbf{f}(\mathbf{x}, m(\tilde{P})) = \mathbf{0}$;
$A \leftarrow J_{\mathbf{x}}\mathbf{f}(\tilde{X}, m(\tilde{P}))$;
$\mathbf{Y} \leftarrow [m(A)]^{-1}$;
repeat
  $\tilde{X}^{\text{old}} = \tilde{X}$;
  $A \leftarrow J_{\mathbf{x}}\mathbf{f}(\tilde{X}, m(\tilde{P}))$, $B \leftarrow -J_{\mathbf{p}}\mathbf{f}(\tilde{X}, \tilde{P})$;
  $A \leftarrow \mathbf{Y}A$, $B \leftarrow \mathbf{Y}B$;
  for $i = 1, \ldots, n_p$ **do**
    $S_i \leftarrow \Gamma(A, B_i, S_i)$; `// uses interval Gauss-Seidel`
  $\tilde{X} \leftarrow \tilde{X} \cap \mathbf{x} + S(\tilde{P} - m(\tilde{P}))$;
until $d_H(\tilde{X}, \tilde{X}^{old}) < \varepsilon$;
return $(\tilde{X}, S)$;

5.4 Case studies

In this section, the convergence order of the sensitivity based bounding method is studied numerically. To this effect, the method is implemented in MATLAB relying on the interval arithmetic provided by INTLAB [146]. Additionally, parametric interval Newton methods are implemented to provide a comparison.

Consider a sequence $\{P^l\} \subset \mathbb{IP}$ and construct $X^l \equiv X^*(P^l)$ using the method proposed in Section 5.3. We compare X^l with $\mathbf{x}(P^l)$, the image of P^l under $\mathbf{x}$, in order to validate the convergence order of the method numerically.

In each case study, let $P^1 = P$ and $P_i^l = m(P_i^{l-1}) + \frac{1}{4}[-w(P_i^{l-1}), w(P_i^{l-1})]$, $i = 1, \ldots, n_p$ and $l = 2, \ldots, 15$. If $\mathbf{x}$ is not known analytically, the image of P^l under $\mathbf{x}$ was numerically estimated using BARON [150] in GAMS by minimizing or maximizing z_i while requiring $(\mathbf{z}, \mathbf{p}) \in X \times P^l$ and $\mathbf{f}(\mathbf{z}, \mathbf{p}) = \mathbf{0}$ for $i = 1, \ldots, n_x$. $X^*(P^l)$ denotes the approximate limit of each bounding method on P^l, which was calculated by stopping the iteration when $d_H(X^{k+1}(P^l), X^k(P^l)) < \varepsilon = 10^{-8}$. Each inclusion function was evaluated using its natural interval extension. $\mathbf{Y}(X, P^1) = [m(J_{\mathbf{x}}(X, P^1))]^{-1}$ and $\mathbf{Y}(X, P^l) = [m(J_{\mathbf{x}}(X^*(P^{l-1}), P^l))]^{-1}$, $l > 1$ were used as preconditioners.

Example 5.2. [163, Example 1] Let $n_x = 2$ and $n_p = 2$. Let $D_x \times D_p = \mathbb{R}^2 \times \mathbb{R}^2$ and consider

$$\mathbf{f}(\mathbf{z}, \mathbf{p}) = \begin{bmatrix} z_1^2 + z_2^2 + p_1 z_1 + 4 \\ z_1 + p_2 z_2 \end{bmatrix}.$$

Let $X = [-1.5, 0] \times [0, 0.5]$ and $P = [5, 7]^2$. In this case, it is possible to construct the implicit function $\mathbf{x} : P \to X$ as

$$\mathbf{x}(\mathbf{p}) = \begin{bmatrix} \frac{p_2\sqrt{p_1^2p_2^2 - 16p_2^2 - 16} - p_1p_2^2}{2(p_2^2+1)} \\ \frac{1}{2}\left(\frac{p_1p_2}{p_2^2+1} - \frac{\sqrt{p_1^2p_2^2 - 16p_2^2 - 16}}{p_2^2+1}\right) \end{bmatrix}.$$

In Figure 5.2, $d_H(X^l, \mathbf{x}(P^l))$ is plotted against $w(P^l)$ validating the second-order convergence.

Example 5.3. [100, Example 4.1] Let $n_x = 3$ and $n_p = 2$. Let $D_x \times D_p = \mathbb{R}^3 \times (\mathbb{R} - \{0\})^2$ and consider

$$\mathbf{f}(\mathbf{z}, \mathbf{p}) = \begin{bmatrix} \frac{3.25 - z_1}{p_1} - z_3 \\ \frac{x_1}{p_2} - z_3 \\ z_2 - \frac{z_1^2}{1+z_1^2} \end{bmatrix}.$$

Let $X = [-30, 30]^2$ and $P = [1800, 2200] \times [900, 1100]$. In this case, it is possible to construct

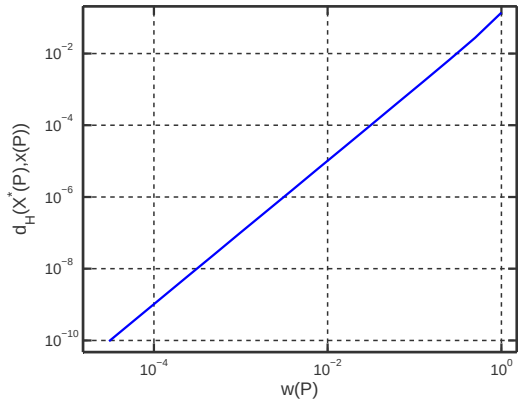


Figure 5.2: Empirical convergence order of the sensitivity based bounding method for Example 5.2

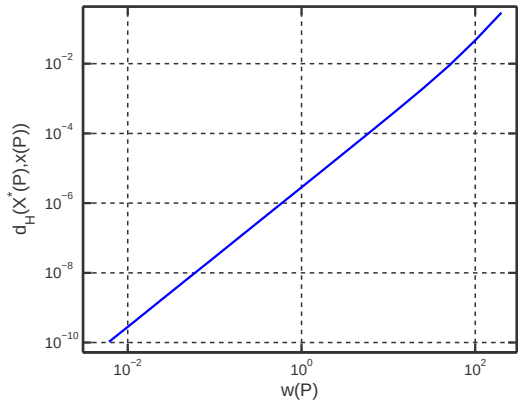


Figure 5.3: Empirical convergence order of the sensitivity based bounding method for Example 5.3

the implicit function $\mathbf{x} : P \to X$ as

$$\mathbf{x}(\mathbf{p}) = \begin{bmatrix} \frac{3.25p_2}{p_1+p_2} \\ \frac{169p_2^2}{16p_1^2+32p_1p_2+185p_2^2} \\ \frac{3.25}{p_1+p_2} \end{bmatrix}.$$

In Figure 5.3, $d_H(X^l, \mathbf{x}(P^l))$ is plotted against $w(P^l)$ validating the second-order convergence.

Example 5.4. [162, Example 4.5.3] Reconsider Example 5.1. In Figure 5.4, $d_H(X^l, \mathbf{x}(P^l))$ is plotted against $w(P^l)$ validating the second-order convergence. Here, we also show the rate of convergence of the parametric Hansen-Sengupta operator and the parametric Krawczyk operator initialized with X. While at the initial box, the overestimation is comparable, the

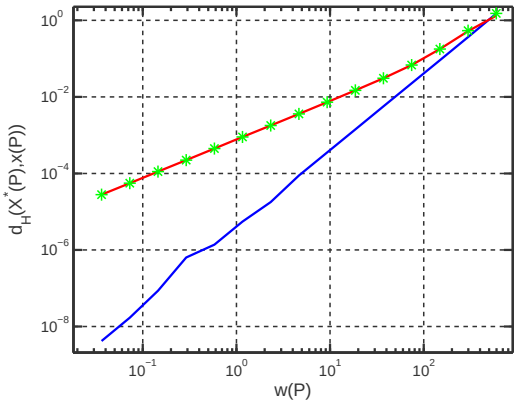


Figure 5.4: Empirical convergence order of the sensitivity based bounding method (blue line), the Hansen-Sengupta operator (red line) and the Krawczyk operator (green asterisks) for Example 5.4

rate of convergence of the latter two methods is less than quadratic.

Example 5.5. [162, Example 4.5.3] Expanding on Example 5.4, we now consider 5 discretization elements. Let $n_x = 25$ and $n_p = 3$. Let $D_x \times D_p = \mathbb{R}^{25} \times \mathbb{R}^3$ and consider

$$\mathbf{f}(\mathbf{z},\mathbf{p}) = \begin{bmatrix} \vdots \\ z_{5(i-1)+1} - z_{5i+1} + \Delta t \left(k_1 z_{5i+4} z_{5i+5} - c_{O_2}(p_1+p_2) z_{5i+1} + \frac{p_1}{K_2} z_{5i+3} + \right. \\ \left. \frac{p_2}{K_3} z_{5i+2} - k_5 z_{5i+1}^2 \right) \\ z_{5(i-1)+2} - z_{5i+2} + \Delta t \left(p_2 c_{O_2} z_{5i+1} - \left(\frac{p_2}{K_3} + p_3 \right) z_{5i+2} \right) \\ z_{5(i-1)+3} - z_{5i+3} + \Delta t \left(p_1 c_{O_2} z_{5i+1} - \frac{p_1}{K_2} z_{5i+3} \right) \\ z_{5(i-1)+4} - z_{5i+4} + \Delta t \left(-k_{1s} z_{5i+4} z_{5i+5} \right) \\ z_{5(i-1)+5} - z_{5i+5} + \Delta t \left(-k_1 z_{5i+4} z_{5i+5} \right) \\ \vdots \end{bmatrix}$$

where $i = 1, \ldots, 5$, $\Delta t = 0.01$, $K_2 = T = 273$, $K_2 = 46 \exp(\frac{6500}{T} - 18)$, $K_3 = 2K_2$, $k_1 = 53$, $k_{1s} = 10^{-6} k_1$, $k_5 = 0.0012$ and $c_{O_2} = 0.02$. Let $X = ([0,140]^3 \times [0,0.4] \times [0,140])^5$ and $P = [10,1200]^2 \times [0.001,40]$. Set $[z_{-4}, \ldots, z_0] = [0,0,0,0.4,140]$.

As demonstrated in [162, p. 128f], the structure of this problem can be exploited easily.

In Figure 5.5, $d_H(X^l, \mathbf{x}(P^l))$ is plotted against $w(P^l)$ validating the second-order convergence. Here, we also show the rate of convergence of the parametric Hansen-Sengupta operator and the parametric Krawczyk operator initialized with X. While at the initial box, the overestimation is comparable, the rate of convergence of the latter two methods is less than quadratic.

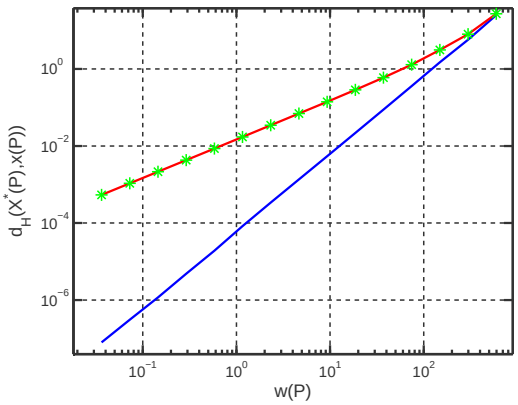


Figure 5.5: Empirical convergence order of the sensitivity based bounding method (blue line), the Hansen-Sengupta operator (red line) and the Krawczyk operator (green asterisks) for Example 5.5

5.5 Conclusion

In this paper, it was argued that the convergence order of the parametric Krawczyk method for bounding the zeros of systems of nonlinear equations is linear in the parameters only. Since second-order convergent bounds are necessary to overcome the cluster problem in global optimization, a method based on the sensitivities of the systems of nonlinear equations was studied. The estimate of the sensitivities were shown to converge linearly so that a centered form of the implicit function is guaranteed to have second-order convergence. Case studies validate that this method does indeed converge more rapidly than the parametric Krawczyk method or the parametric Hansen-Sengupta operator.

Chapter 6

Global optimization of bounded factorable functions with discontinuities

Recently, McCormick's method [118], which was originally defined for explicitly stated factorable functions, has been extended to continuous functions that are not known explicitly, e.g., when they result from algorithms [121, 156]. Discontinuities can appear, for example, in algorithms with conditional statements (i.e., `IF-THEN-ELSE`), which have been excluded in a previous paper [121, p. 593]. Hereafter, a class of *discontinuous* functions is considered. Neither are the standard relaxation techniques, which are defined for continuous functions, applicable in this case nor has it been considered in branch-and-bound theory [88].

A closer look at McCormick's composition theorem and its proof [118], however, indicates that the result can be extended to discontinuous factorable functions if they are bounded. It is, in addition, necessary to know valid relaxations of univariate discontinuous functions. Since a discontinuity can be represented by a step function [182], for which relaxations can be constructed easily, this requirement can be met when the factorable representation of the function has a finite number of discontinuous univariate factors. However, it is not clear if the properties of McCormick relaxations shown in [156] hold true. In this chapter, the obtained relaxations for bounded factorable functions with discontinuities are analyzed in detail. Such analysis is indispensable in order to establish properties of the proposed relaxation technique required for its use in a branch-and-bound method [88]. Furthermore, branch-and-bound theory must be extended as continuity is a standing assumption throughout [88].

The proposed method is particularly well suited to solve optimization problems with discontinuities depending on continuous variables. Examples of this case are discontinuous cost functions in process design: when a certain size is exceeded, two units need to be used instead of one, causing a discontinuity in the investment cost (this starkly contrasts with discrete decisions that require integer variables, e.g., when two exclusive alternatives for one unit exist). Examples for such problems can be found in process synthesis with discontinuous investment costs [168] as well as dynamic optimization problems with discontinuities [14], in particular, hybrid systems [112]. Currently, mixed-integer or complementarity constraint formulations are often used to model discontinuities depending on continuous variables [20, 168]. In the former, binary variables are introduced to model discontinuities whereas in the latter complementarity constraints take on this role. Commercial global optimization algorithms are available for MINLPs [150], however,

introducing binary variable to model the discontinuities can increase the number of variables drastically. This can lead to poor performance as branch-and-bound algorithms are known to scale worst-case exponentially with the number of variables. MPECs are usually reformulated as NLPs and are only solved locally at present [20, 61].

While this chapter focuses on *global* optimization of discontinuous factorable functions, it should be noted that existing work on discontinuous optimization considered finding *locally* optimal solutions. Aside from using the definition of a local minimum as a point attaining the smallest value of the objective function in a neighborhood, no other characterization, e.g., using gradients, is applicable when the function is not even continuous. In the quest to derive local optimality conditions, the notion of derivatives is generalized to nonsmooth and discontinuous functions by several authors [15, 16, 45, 58, 124, 183]. Recently, Rockafellar generalized derivatives have been used in direct search algorithms for discontinuous functions [172] . Another prevalent idea in the literature is to approximate the discontinuous function by convolving it with an appropriate mollifier resulting in an averaged function. This operation leads to an integration problem, possibly of high dimension, which is computationally expensive and is often evaluated using Monte Carlo schemes [17, 57, 144, 183]. Conn and Mongeau [47] consider piecewise linear optimization problems where the objective function and constraints have discontinuities on a set of hyperplanes and propose an algorithm to identify local minima. In an approach more closely related to the idea proposed in this chapter Zang [182] introduces step functions to express the discontinuities and suggests a family of smooth approximations for these. Similar ideas are used to smooth continuous functions at points of non-differentiability [e.g., 55] and are prone to introduce inaccuracy and numerical instability.

Definition 6.1 ([cf. 175, p. 45])**.** A function $f : D \subset \mathbb{R}^n \to \mathbb{R}$ is called *lower semi-continuous* if

$$\liminf_{\mathbf{x}\to\mathbf{x}_0} f(\mathbf{x}) \geq f(\mathbf{x}_0), \quad \forall \mathbf{x}_0 \in D$$

where $\liminf_{\mathbf{x}\to\mathbf{x}_0} f(\mathbf{x}) = \lim_{\delta\to 0} \inf\{f(\mathbf{x}) : \mathbf{x} \in X, 0 < \|\mathbf{x} - \mathbf{x}_0\|_2 < \delta\}$.

It is well-known that lower semi-continuity of a function f is a sufficient property for f to attain its infimum on a nonempty compact set; cf. [82].

Theorem 6.1. *Suppose $D \subset \mathbb{R}^n$ nonempty and compact. If a function $f : D \to \mathbb{R}$ is lower semi-continuous on D, then f is bounded from below and it attains its minimum.*

Remark 6.1. When the assumption of lower semi-continuity of f is replaced by boundedness of f from below, one may still obtain a sequence converging to the infimum. Furthermore, the method presented in this chapter will be able, as will be argued in Remark 6.6 and in Section 6.2, to construct such a sequence for optimization problems involving factorable functions.

The remainder of this chapter is organized as follows. First, McCormick relaxations [118] are studied taking advantage of the formalization provided by [156]. In Section 6.1, properties such as continuity and convexity of the obtained relaxations of certain bounded functions are proved. Examples of the relaxations of discontinuous functions are provided.

Furthermore, the behavior of the relaxations on sequences of intervals is investigated, after the necessary assumptions required for these properties are made explicit. This leads up to the results in Section 6.2, where it is argued that a branch-and-bound algorithm with finite ε-convergence can be constructed. The chapter continues with some examples in Section 6.3 which showcase the numerical feasibility and provide first promising examples from different applications. The chapter is concluded with a summary of the obtained results in Section 6.4.

6.1 Relaxations of bounded $\mathcal{L}$-factorable functions

In this section, the construction of relaxations for factorable and bounded functions is discussed. To this end, the well-known results for obtain McCormick relaxations [118, 156] are extended. Then, discontinuous univariate intrinsic functions are studied more closely and first examples of the constructed relaxations are given. A discussion about the behavior of the relaxations on sequences of intervals is preceded by a collection of necessary assumptions and concludes this section.

Here, the notions of $\mathcal{L}$-computational sequences and $\mathcal{L}$-factorable functions will be extended by only requiring boundedness, but not continuity of the univariate functions in $\mathcal{L}$. While it was not assumed explicitly, Assumptions 3.2 and 3.4 stating that, for any $(u, B, \mathbb{IR}) \in \mathcal{L}$, the corresponding interval extension and McCormick extension of u is locally Lipschitz on $\mathbb{I}B$ and $\mathbb{M}B$, respectively, also guaranteed continuity of the real-valued univariate function on B. If we drop these assumptions, the resulting class of computational sequences and functions are more general and will be called *bounded $\mathcal{L}$-computational sequences* and *bounded $\mathcal{L}$-factorable functions*. The class of bounded $\mathcal{L}$-factorable functions includes most functions that can be represented finitely on a computer.

In the literature [e.g., 118, 121, 155, 156], a standing assumption is continuity of the univariate functions $(u, B, \mathbb{R}) \in \mathcal{L}$ and hence f. When C is compact, continuity of each operation in Definition 3.2 guarantees compactness of $f(C)$ [145]. Hence, continuous factorable functions are always bounded factorable on a compact set X. As shown below, if each univariate function is bounded, then the constructed function is bounded factorable.

Lemma 6.1. *Suppose $D \subset \mathbb{R}^n$ is bounded. Consider a $\mathcal{L}$-factorable function $f : D \to \mathbb{R}$. f is bounded $\mathcal{L}$-factorable if each univariate $(u, B, \mathbb{R}) \in \mathcal{L}$ is bounded on B.*

Proof. For $1 \leq k \leq n$, the assertion holds trivially. Suppose the assertion holds for some k where $n < k \leq n_f$. When v_k is defined by Definition 3.2 (a), v_k is certainly bounded. When v_k is defined by Definition 3.2 (b), v_k is bounded since o_k is bounded. From finite induction, it follows that v_{n_f} is bounded and, hence, f is bounded $\mathcal{L}$-factorable. □

6.1.1 Extension of McCormick's result to bounded $\mathcal{L}$-factorable functions

McCormick [118] presented a recursive procedure to create relaxations of factorable functions f on a nonempty convex set D. While in his exposition, McCormick restricted the

result to continuous factorable functions, it can be easily extended to bounded factorable functions.

Theorem 6.2. *Let $D \subset \mathbb{R}^n$ be a nonempty convex set. Consider the composite function $f = f_2 \circ f_1$ where $f_1 : D \to \mathbb{R}$ is bounded on D, let $f_1(D) \subset [a,b]$ and $f_2 : [a,b] \to \mathbb{R}$. Suppose that relaxations $\check{f}_1$ and $\hat{f}_1$ of f_1 on C as well as relaxations $\check{f}_2$ and $\hat{f}_2$ of f_2 on $[a,b]$ are known. Let $z_{\min}$ be a point at which f_2 attains its infimum on $[a,b]$, and let $z_{\max}$ be a point at which $\hat{f}_2$ attains its supremum on $[a,b]$. Then*

$$\check{f}(\mathbf{x}) = \check{f}_2[\mathrm{mid}\{\check{f}_1(\mathbf{x}), \hat{f}_1(\mathbf{x}), z_{\min}\}]$$

is a convex relaxation of $f_2 \circ f_1$ on C, and

$$\hat{f}(\mathbf{x}) = \hat{f}_2[\mathrm{mid}\{\check{f}_1(\mathbf{x}), \hat{f}_1(\mathbf{x}), z_{\max}\}]$$

is a concave relaxation of $f_2 \circ f_1$ on C, where $\mathrm{mid} : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ *selects the middle value of the three scalar arguments.*

Proof. The original proof [119] remains valid after the continuity hypothesis on f_1 is replaced with a boundedness assumption as only $f_1(D) \subset [a,b]$ is needed. □

As demonstrated in Chapter 3, Theorem 6.2 allows the construction of relaxations of complicated functions by decomposing the function into factors for which relaxations are known [118, 156]. A precise definition of this procedure was first given in [156] and is listed in Chapter 3.

Most importantly, allowing for discontinuous but bounded univariate functions in the library $\mathcal{L}$ does not invalidate Theorem 2.4.32 in [155] as long as Assumption 3.3 holds for all elements of $\mathcal{L}$.

Theorem 6.3. *Let $(\mathcal{S}, \pi_o)$ be a bounded $\mathcal{L}$-computational sequence. The natural McCormick extension $(\mathcal{F}_\mathcal{S}, \mathcal{D}_\mathcal{S}, \mathbb{MR}^{n_o})$ is a coherently concave, inclusion monotonic McCormick extension of $(\mathbf{f}_\mathcal{S}, D_\mathcal{S}, \mathbb{R}^{n_o})$.*

Proof. Follows from Theorem 2.4.32 in [155] in conjunction with Theorem 6.2. □

Despite the discontinuity of $\mathbf{f}_\mathcal{S}$, it is still possible to show that the standard McCormick relaxations are continuous given a modification of Assumption 2.5.39 in [155].

Theorem 6.4. *Assume that for each element $u \in \mathcal{L}$ with McCormick extension $(u, \mathbb{M}B, \mathbb{MR})$, $u(Z^B, \cdot)$ is continuous on $\mathbb{I}B$. Then, the standard McCormick relaxations of a bounded $\mathcal{L}$-factorable function f on X are locally Lipschitz on X for any $X \in \mathbb{I}D$.*

Proof. Follows from finite induction using the continuity assumption for each $o_k \in \mathcal{L}$ together with Lemma 2.5.38 in [155], which shows that $(+, \mathbb{MR}^2, \mathbb{MR})$ and $(\times, \mathbb{MR}^2, \mathbb{MR})$ are locally Lipschitz on $\mathbb{MR}^2$. □

6.1.2 Univariate piecewise continuous functions

From Definition 3.2, it is apparent that discontinuities of a bounded $\mathcal{L}$-factorable function must stem from discontinuities in some of elements of $\mathcal{L}$. When there is only a finite number of discontinuities, these functions can be reduced to products with a generic step function, which incorporates the discontinuity, and continuous functions.

Suppose $u : X \to \mathbb{R}$ is of the form

$$u(x) = \begin{cases} \varphi_1(x) & \text{if } x \in [\underline{x}_1, \overline{x}_1], \\ \varphi_2(x) & \text{if } x \in (\underline{x}_2, \overline{x}_2], \end{cases}$$

where $X, X_1, X_2 \in \mathbb{IR}$, $[\underline{x}_1, \overline{x}_1] = X_1$, $(\underline{x}_2, \overline{x}_2] \subset X_2$, $\varphi_1 : X_1 \to \mathbb{R}$, $\varphi_2 : X_2 \to \mathbb{R}$, $X = X_1 \cup X_2$, and $\overline{x}_1 = \underline{x}_2$ and let φ_1, φ_2 be continuous on their respective domains. Denote the step function as $\psi : \mathbb{R} \to \mathbb{R}$, i.e.,

$$\psi(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ 1 & \text{otherwise.} \end{cases}$$

Then $u(x)$ can be represented by

$$u(x) = \psi(x - \overline{x}_1)\varphi_2(x) + [1 - \psi(x - \overline{x}_1)]\varphi_1(x). \tag{6.1}$$

As a result, it is sufficient to analyze only $\psi(x)$ in detail. The following result summarizes the information relevant for the construction of McCormick relaxations.

Theorem 6.5. *Consider $\psi : \mathbb{R} \to \mathbb{R}$ as defined above. Then, $(\Psi, \mathbb{IR}, \mathbb{IR})$ given by*

$$\Psi(X) = \begin{cases} [0,0] & \text{if } \overline{x} \leq 0, \\ [1,1] & \text{if } \underline{x} > 0, \\ [0,1] & \text{otherwise,} \end{cases}$$

satisfies Assumption 3.1 and $\check{\psi}(X,x)$, $\hat{\psi}(X,x)$ given by

$$\check{\psi}(X,x) = \begin{cases} 0 & \text{if } \overline{x} \leq 0 \vee (\overline{x} > 0 \wedge x \leq 0), \\ 1 & \text{if } \underline{x} > 0, \\ x/\overline{x} & \text{otherwise,} \end{cases}$$

and

$$\hat{\psi}(X,x) = \begin{cases} 0 & \text{if } \overline{x} \leq 0, \\ 1 & \text{if } \underline{x} > 0 \vee (\underline{x} \leq 0 \wedge x \geq 0), \\ 1 - x/\underline{x} & \text{otherwise.} \end{cases}$$

as well as $x^{\min}(X) = \underline{x}$, $x^{\max}(X) = \overline{x}$ satisfy Assumption 3.3 for all $X \in \mathbb{IR}$ and $x \in X$. Consequently, a coherently concave, inclusion monotonic McCormick extension of ψ can be obtained as defined by Equation (3.2). Furthermore, $\psi(X^B, \cdot)$ is continuous on $\mathbb{IR}$ for any $X^B \in \mathbb{IR}$.

Proof. It is easy to check the validity of the bounds and the inclusion monotonicity property.

Similarly, the relaxations are easy to check when $0 \notin X$. If $0 \in X$, consider the convex hull of the epigraph of ψ which yields the convex underestimator given in the result. Similarly, the concave overestimator is given by the convex hull of the hypograph.

Theorems 2.4.27, 2.4.29 and 2.4.30 in [155] are sufficient in light of Theorem 6.3 to establish that $(\psi, \mathbb{MR}, \mathbb{MR})$ is a coherently concave, inclusion monotonic McCormick extension.

Lastly, since the mid function is continuous and $\check{\psi}(X^B, \cdot)$ and $\hat{\psi}(X^B, \cdot)$ are continuous on $\mathbb{R}$, it follows that $\psi(X^B, \cdot)$ is continuous on $\mathbb{IR}$. □

Remark 6.2. Strictly, φ_1 and φ_2 are defined on X_1 and X_2 only. When one defines $o_i(x) = +\infty$ for $x \notin X_i$ and $0 \cdot +\infty = 0$, the above statement also holds. Furthermore, when φ_1 is defined on X, an alternative to (6.1) is

$$u(x) = \psi(x - \overline{x}_1)[\varphi_2(x) - \varphi_1(x)] + \varphi_1(x).$$

6.1.3 Examples of constructed relaxations

Next, it is demonstrated how more complicated functions with discontinuities can be expressed using the previously introduced function ψ and the thus computed relaxations are showcased. Example 6.1 shows how to model a function with multiple discontinuities, including a point where the function attains neither its lower nor its upper limit. Example 6.2 demonstrates that the discontinuity can depend on a factorable function of the variables. In each case, the calculations are implemented using `MC++`, the successor of `libMC` [41, 121], enhanced with functionality for ψ.

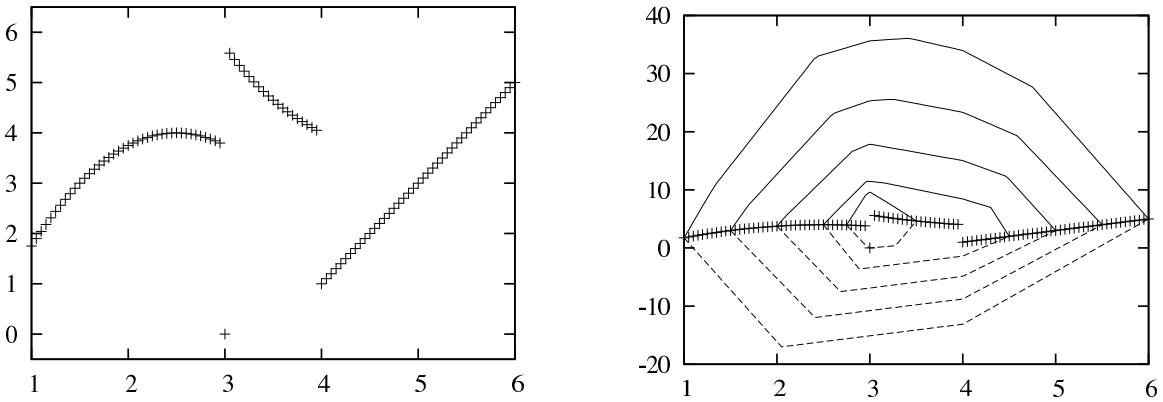
Example 6.1. Consider the lower semi-continuous function $f_1 : [1, 6] \to \mathbb{R}$ with

$$f_1(x) = \begin{cases} -(x - 2.5)^2 + 4 & \text{if } x \in [1, 3), \\ 0 & \text{if } x = 3, \\ e^{4-x} + 3 & \text{if } x \in (3, 4), \\ 2x - 7 & \text{if } x \in [4, 6]. \end{cases}$$

It can be represented as

$$\begin{aligned} f_1(x) =& \psi(4 - x) \left\{ \psi(x - 3) \left[e^{4-x} + 3 - \left\{ \psi(3 - x)(-(x - 2.5)^2 + 4 - 0) + 0 \right\} \right] \right. \\ & \left. + \left[\psi(3 - x)(-(x - 2.5)^2 + 4 - 0) + 0 \right] - (2x - 7) \right\} + (2x - 7). \end{aligned}$$

Its graph and a selection of the constructed relaxations are showcased in Figure 6.1. It is worth while to point out several observations. First, the example shows that it is possible to model functions with multiple discontinuities, including such where the function does not attain either one-sided limit. Second, the generated relaxations are generally nonsmooth. This is characteristic for McCormick relaxations and has been noted previously [121].



(a) Graph of f_1

(b) Graph of f_1 and its relaxations

Figure 6.1: Graph of f_1 (indicated by $+$) as well as convex relaxations (dashed line) and concave relaxations (continuous line) constructed on several intervals. Note that the scales on the vertical axes differ.

Example 6.2. Consider the lower semi-continuous function $f_2 : [0.5, 1.5]^2 \to \mathbb{R}$ with

$$f_2(x,y) = \begin{cases} 0.5\sin(6y-1)x^2 & \text{if } xy > 1, \\ 2(x+y) - e^{xy+1} & \text{if } xy \leq 1. \end{cases}$$

It can be represented as

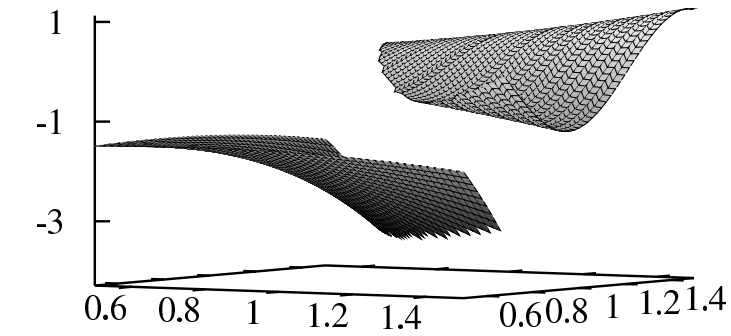
$$f_2(x,y) = \psi(1-xy)\left[2(x+y) - e^{xy+1} - 0.5\sin(6y-1)x^2\right] + 0.5\sin(6y-1)x^2.$$

Its graph and a selection of the constructed relaxations are showcased in Figure 6.2. Note that ψ can take any arbitrary factor as argument, in this case a bilinear term, and thus the discontinuity can depend on the variables nonlinearly.

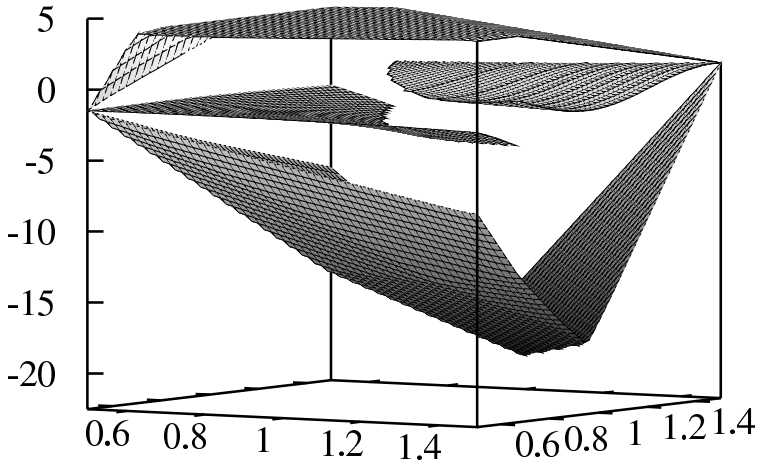
6.1.4 Assumptions on f, o_k and the interval and McCormick extension of o_k

In Section 6.1.6, the convergence properties of standard McCormick relaxations of bounded factorable functions will be investigated. Prior to this, some assumptions about the interval extensions and the relaxations of the univariate functions will be made. This approach allows a more general discussion compared to only studying a selection of univariate intrinsic functions or particular factorable functions.

In addition to Assumptions 3.1 and 3.3, two additional assumptions will be made subsequently. While the previous assumptions have also been introduced in [155] and Assumption 6.1 is a less stringent assumption than Assumption 2.5.39 in [155], Assumption 6.2 is newly introduced here and will be discussed in more detail below. In the setting considered in [155], it can be taken for granted.



(a) Graph of f_2



(b) Graph of f_2 and its relaxations

Figure 6.2: Graph of f_2 as well as its convex and concave relaxations on $[0.5, 1.5]^2$. Note that the scales on the vertical axes differ.

Assumption 6.1. Assume that for each $u \in \mathcal{L}$ with McCormick extension $(u, \mathbb{M}B, \mathbb{M}\mathbb{R})$, $u(Z^B, \cdot)$ is continuous on $\mathbb{I}B$ for any $Z^B \in \mathbb{I}B$.

Note that is has been shown in Theorem 6.5 that the convex and concave relaxations of ψ satisfy Assumption 6.1.

In order to streamline the presentation, the next assumption will be introduced, which is sufficient to prove convergence of f to f. This assumption is discussed in more detail in Section 6.1.5. There, more insight into prerequisites for convergence of the relaxations to the function is given. Lastly, it should be pointed out that this assumption is imposed on a given factorization of a bounded factorable function f while the previous assumptions were imposed on $\mathcal{L}$.

Assumption 6.2. Consider a nested sequence of intervals $X^l \to X^* = [\mathbf{x}^*, \mathbf{x}^*]$, $X^l \in \mathfrak{D}_S$,

i	v_i	V_i^l	$\check{v}_i^l$	$\hat{v}_i^l$
1	x	$[-l^{-1}, l^{-1}]$	x	x
2	$\psi(v_1)$	$[0,1]$	$\begin{cases} 0 & \text{if } x \leq 0 \\ xl & \text{otherwise} \end{cases}$	$\begin{cases} 1+xl & \text{if } x \leq 0 \\ 1 & \text{otherwise} \end{cases}$
3	$v_2 - v_2$	$[-1,1]$	$\begin{cases} -xl-1 & \text{if } x \leq 0 \\ xl-1 & \text{otherwise} \end{cases}$	$\begin{cases} 1+xl & \text{if } x \leq 0 \\ 1-xl & \text{otherwise} \end{cases}$

Table 6.1: Factorization of $f = \psi(x) - \psi(x)$ on $X^l = [-l^{-1}, l^{-1}]$.

$X^l \neq X^*$. For each $n_i < k \leq n_f$, assume that for each $n < k \leq m$

$$\lim_{l\to\infty} V_k(X^l) = [\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} v_k(\mathbf{x}), \lim_{l\to\infty} \sup_{\mathbf{x}\in X^l} v_k(\mathbf{x})]. \tag{6.2}$$

Note that $\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} v_k(\mathbf{x})$ does *not* refer to $\liminf_{\mathbf{x}\to\mathbf{x}^*} v_k(\mathbf{x})$. Assumption 6.2 states that, as X^l approaches the degenerate interval X^*, the bounds computed for each of the factors become as tight as possible. Since the bounds on the step mappings v_k are obtained from interval arithmetic, this clearly holds when f is composed of continuous factors. When f is discontinuous however, this is not necessarily true. For example, the dependency problem in interval arithmetic is exacerbated and bounds do not necessarily converge to the function as the host set converges to a degenerate interval. This is demonstrated in the example below.

Example 6.3. Consider the *continuous* function $f : [-1,1] \to \mathbb{R}$ with $f(x) = \psi(x) - \psi(x)$. It can be equivalently written as $f(x) = 0$. Consider the nested sequence of intervals $X^l = [-l^{-1}, l^{-1}]$ that converges to $X^* = [x^*, x^*]$ with $x^* = 0$. It can be shown that the relaxations do not converge in this case. Consider this factorization given in Table 6.1. For all l, the relaxations of f constructed on X^l evaluated at x^* yield $\check{v}_3^l(X^l, [0,0]) = -1$ and $\hat{v}_3^l(X^l, [0,0]) = 1$, i.e., $\lim_{l\to\infty} \check{v}_3^l(X^l, [0,0]) = -1$ and $\lim_{l\to\infty} \hat{v}_3^l(X^l, [0,0]) = 1$ whereas the relaxations of f constructed on the degenerate interval X^* are $\check{v}_3^*(X^l, [0,0]) = f(0) = 0$ and $\hat{v}_3^*(X^l, [0,0]) = f(0) = 0$, also see Figure 6.3.

Note that, in this case, there exists a factorization that circumvents this dependency problem, namely $f(x) = 0$. Thus, depending on the problem formulation, this limitation may be avoided.

Similarly, applying a univariate function to a discontinuous factor may lead to bounds that do not converge to the infimum/supremum as $X^l \to X^*$. To see this, consider the univariate function $u(x) = \cos(x - 0.75)$, the bounded factorable function $f(x) = u(\psi(x))$ and $X^l = [-l^{-1}, l^{-1}]$, $X^* = [0,0]$. Again, $\Psi(X^l) = [0,1]$ and $F(X^l) = [\cos(-0.75), 1]$ while $\Psi(X^*) = [0,0]$ and $F(X^*) = [\cos(-0.75), \cos(-0.75)]$. Furthermore, $f(x) = \cos(-0.75)$ for all $x \in [-1,0]$ and $f(x) = \cos(0.25)$ for all $x \in (0,1]$. Thus, the upper bound does not converge to the supremum as desired. This is due to the fact that there exists a $y \in (0,1)$ so that $u(y) > \max(u(0), u(1))$. Again, this can be avoided when the problem is recast as $f(x) = \psi(x)(\cos(0.25) - \cos(-0.75)) + \cos(-0.75)$. Similar examples can be constructed

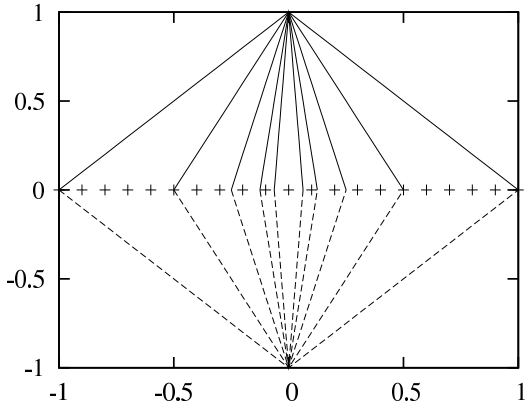


Figure 6.3: Graph of f (indicated by $+$) as well as five of its convex and concave relaxations (indicated by dashed and continuous lines, respectively) for $l = 1, 2, 4, 8, 16$.

so that the lower bound does not converge to the infimum.

6.1.5 Discussion of sufficient conditions for convergence of the relaxations

Here, three lemmata will be given that present sufficient conditions for Assumption 6.2 to hold for a given factor v_k and thus can be used in a finite induction argument to establish Assumption 6.2. In particular, they formalize the discussion in Section 6.1.4 and show that, to establish Assumption 6.2, it is sufficient to exclude these cases from occurring. First, overestimation in binary operations is considered. Here, two reasonably strong results can be given. Then, attention will be directed to univariate functions where more restrictive assumptions need to be made.

Lemma 6.2. *Consider any k such that $n_i < k \leq n_f$ where v_k is defined by a summation or multiplication. Consider a nested sequence of intervals $X^l \to X^* = [\mathbf{x}^*, \mathbf{x}^*]$, $X^l \in \mathbb{IR}$, and $X^l \neq X^*$. Suppose Assumption 6.2 holds for all $i, j < k$. Suppose that v_i and v_j are discontinuous with respect to $\mathbf{x}$ at $\mathbf{x}^*$ and that these discontinuities are introduced at earlier factors $k_i \leq i$ and $k_j \leq j$, i.e., $v_{k_i} = \psi \circ \pi_{k_i}(v_1, \ldots, v_{k_i - 1})$ and $v_{k_j} = \psi \circ \pi_{k_j}(v_1, \ldots, v_{k_j - 1})$. Assume that v_{k_i} and v_{k_j} are the only discontinuous elements. Define subsets of X^l as $\Xi_i^l = \{\mathbf{x} \in X^l : \pi_{k_i}(v_1(\mathbf{x}), \ldots, v_{k_i}(\mathbf{x})) > 0\}$ and $\Xi_j^l = \{\mathbf{x} \in X^l : \pi_{k_j}(v_1(\mathbf{x}), \ldots, v_{k_j - 1}(\mathbf{x})) > 0\}$. If there exists a $L \in \mathbb{N}$ so that for all $l > L$,*

$$\Xi_i^l \cap \Xi_j^l \neq \emptyset, \quad \Xi_i^l \cap (X^l \backslash \Xi_j^l) \neq \emptyset, \quad (X^l \backslash \Xi_i^l) \cap \Xi_j^l \neq \emptyset, \quad (X^l \backslash \Xi_i^l) \cap (X^l \backslash \Xi_j^l) \neq \emptyset,$$

then Assumption 6.2 holds for k.

Proof. By assumption, there exist four sequences $\{\mathbf{x}_1^l\}, \ldots, \{\mathbf{x}_4^l\}$ converging to $\mathbf{x}^*$ where $\mathbf{x}_1^l \in \Xi_i^l \cap \Xi_j^l$, $\mathbf{x}_2^l \in \Xi_i^l \cap (X^l \backslash \Xi_j^l)$, $\mathbf{x}_3^l \in (X^l \backslash \Xi_i^l) \cap \Xi_j^l$, and $\mathbf{x}_4^l \in (X^l \backslash \Xi_i^l) \cap (X^l \backslash \Xi_j^l)$.

For any X^l with $l > L$, the image of v_{k_i} is $V_{k_i}^l = [0, 1]$ as Ξ_i^l is a nonempty strict subset of X^l, $v_{k_i}(\mathbf{x}_q^l) = 1$ for $q = 1, 2$ and $v_{k_i}(\mathbf{x}_q^l) = 0$ for $q = 3, 4$. Thus, $V_{k_i}^l$ is an exact bound of

the range of v_{k_i}. Consider the finite sequence of $s+1$ continuous factors, say $v_{i_1}, \ldots, v_{i_s}, v_i$ with $k_i < i_1 < \ldots < i_s < i$, that maps V_{k_i} to V_i. By assumption, other arguments involved in the definition of the factors $v_{i_1}, \ldots, v_{i_s}, v_i$ are continuous step mappings and, as a result, their corresponding interval bounds converge to degenerate intervals as $l \to \infty$.

Consider factor v_{i_1} and let $[\underline{v}^*_{i_1}, \overline{v}^*_{i_1}] = \lim_{l\to\infty}[\underline{v}^l_{i_1}, \overline{v}^l_{i_1}]$. If this step mapping is a binary operation combining v_{k_i} with a continuous factor, V_{i_1} will converge to a non-degenerate interval and, without loss of generality, $\underline{v}^*_{i_1} = \lim_{l\to\infty} v^l_{i_1}(\mathbf{x}^l_q)$ for $q = 1,2$ and $\overline{v}^*_{i_1} = \lim_{l\to\infty} v^l_{i_1}(\mathbf{x}^l_q)$ for $q = 3,4$. If this step mapping is a univariate operation, Assumption 6.2 guarantees that V_{i_1} will converge to the exact bounds, i.e., without loss of generality, $\underline{v}^*_{i_1} = \lim_{l\to\infty} v^l_{i_1}(\mathbf{x}^l_q)$ for $q = 1,2$ and $\overline{v}^*_{i_1} = \lim_{l\to\infty} v^l_{i_1}(\mathbf{x}^l_q)$ for $q = 3,4$. Repeating this argument for the factors $v_{i_2}, \ldots, v_{i_s}, v_i$, it follows without loss of generality that $\underline{v}^*_i = \lim_{l\to\infty} v^l_i(\mathbf{x}^l_q)$ for $q = 1,2$ and $\overline{v}^*_i = \lim_{l\to\infty} v^l_i(\mathbf{x}^l_q)$ for $q = 3,4$ where $[\underline{v}^*_i, \overline{v}^*_i] = \lim_{l\to\infty}[\underline{v}^l_i, \overline{v}^l_i]$. It can be argued similarly that, without loss of generality, $\underline{v}^*_j = \lim_{l\to\infty} v^l_j(\mathbf{x}^l_q)$ for $q = 1,3$ and $\overline{v}^*_j = \lim_{l\to\infty} v^l_j(\mathbf{x}^l_q)$ for $q = 2,4$ where $[\underline{v}^*_j, \overline{v}^*_j] = \lim_{l\to\infty}[\underline{v}^l_j, \overline{v}^l_j]$.

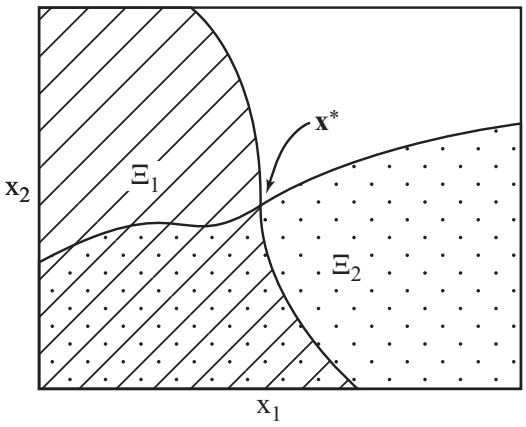
Thus, each combination of the bounds of v_i and v_j is attained in the neighborhood of $\mathbf{x}^*$. In particular, in the case of addition, the sequences $\{v_i(\mathbf{x}^l_1)\}$, $\{v_j(\mathbf{x}^l_1)\}$ and $\{v_i(\mathbf{x}^l_4)\}$, $\{v_j(\mathbf{x}^l_4)\}$ converge to $\underline{v}^*_i$, $\underline{v}^*_j$ and $\overline{v}^*_i$, $\overline{v}^*_j$, respectively. Thus, $[\underline{v}^*_k, \overline{v}^*_k] = [\underline{v}^*_i, \overline{v}^*_i] + [\underline{v}^*_j, \overline{v}^*_j]$ is, in the limit, an exact bound. A similar argument can be presented for the case of multiplication. Here, each combination of lower and upper bounds on v_i and v_j is realized by a different sequence $\{\mathbf{x}^l_q\}$, $q = 1, \ldots, 4$. Thus, Assumption 6.2 holds for k. □

Remark 6.3.

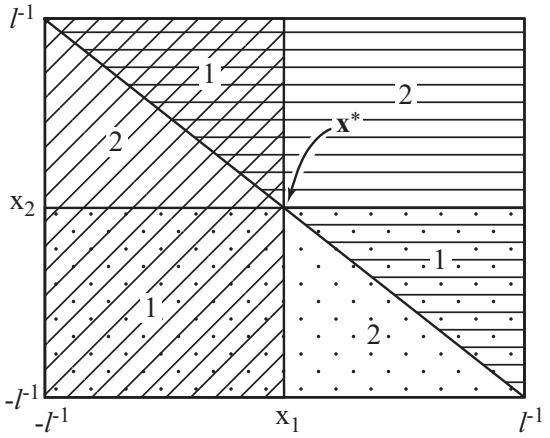
- Lemma 6.2 considers the case of adding or multiplying v_i and v_j where v_i and v_j are discontinuous in the limit $\mathbf{x}^*$ and these discontinuities are introduced by exactly one ψ function each. Then, the dependency problem in interval arithmetic can be mitigated when there exist regions in each interval X^l so that all combination of the lower and upper bounds of the factors v_i and v_j are attained. This can be alternatively expressed as requiring that the intrinsic discontinuities do not coincide in a neighborhood of $\mathbf{x}^*$. A case where this hypothesis of Lemma 6.2 holds is illustrated in Figure 6.4 (a).

- A counterexample can be given to show that Lemma 6.2 cannot be easily extended to the case when more than n intrinsic discontinuities coincide at $\mathbf{x}^* \in \mathbb{R}^n$. To see this, consider $f(\mathbf{x}) = 1 + \psi(x_1) + \psi(x_2) - \psi(x_1 + x_2)$, $X = [-1,1]^2$ and $X^l = [-l^{-1}, l^{-1}]^2$. As shown in Figure 6.4 (b), three intrinsic discontinuities coincide at $(0,0)$. The bounds of f on X^l obtained from the natural interval extension of f are $F(X^l) = [0,3]$. They are not attained for any $\mathbf{x} \in X^l$ and any l and thus Assumption 6.2 does not hold.

- Also note that, given Assumption 6.2, the exacerbated dependency problem of interval arithmetic is not acute when there is only one discontinuity present in either v_i or v_j at $\mathbf{x}^*$. This has been exploited in the proof of Lemma 6.2.



(a) Illustration showing a case where hypothesis of Lemma 6.2 holds



(b) Counterexample showing when Assumption 6.2 does not hold. The digits indicate the value of $f(\mathbf{x}) = 1 + \psi(x_1) + \psi(x_2) - \psi(x_1 + x_2)$ in the 6 subsets of $X^l = [-l^{-1}, l^{-1}]^2$, note that $\underline{f} = 0$, $\overline{f} = 3$.

Figure 6.4: Illustrations for Assumption 6.2 when $X \subset \mathbb{R}^2$. The curves indicate discontinuities introduced at previous factors.

- Lastly, observe that the hypotheses of Lemma 6.2 cannot be satisfied when $X^l \subset \mathbb{R}$. At most three subsets of X in the vicinity of x^*, $\{x : x < x^*\}$, $\{x : x = x^*\}$ and $\{x : x > x^*\}$, are conceivable where v_i and v_j could attain their lower and upper bounds. To guarantee that Assumption 6.2 holds for v_k, the interval arithmetic for $v_i + v_j$ or $v_i v_j$ needs to combine the bounds in such a way that v_k attains both its lower and upper bound. However, it is easy to conceive counterexamples where this is not true, e.g., see the discussion prior to Lemma 6.2.

Though it was pointed out that there are counterexamples restricting the generalization of Lemma 6.2 when more than 2 intrinsic discontinuities coincide at $\mathbf{x}^*$ in $\mathbb{R}^2$, a generalization is possible to n intrinsic discontinuities coinciding in $\mathbb{R}^n$.

Lemma 6.3. *Consider any k such that $n < k \leq m$ where v_k is defined by summation or multiplication. Suppose Assumption 6.2 holds for all $i, j < k$. Consider a nested sequence of intervals $X^l \to X^* = [\mathbf{x}^*, \mathbf{x}^*]$, $X^l \in \mathbb{IC}$, $X^l \neq X^*$. Suppose that v_i and v_j are discontinuous with respect to $\mathbf{x}$ at $\mathbf{x}^*$ and that these discontinuities are introduced by $q \leq n$ earlier factors $k_1, \ldots, k_q$, i.e., $v_{k_{\tilde{q}}} = \psi \circ \pi_{k_{\tilde{q}}}(v_1, \ldots, v_{r_{\tilde{q}}})$ with $v_{r_{\tilde{q}}}(\mathbf{x}^*) = 0$ for $\tilde{q} = 1, \ldots, q$. Assume that $v_{r_{\tilde{q}}}$ is differentiable with respect to $\mathbf{x}$ at $\mathbf{x}^*$, for all $\tilde{q} = 1, \ldots, q$, and denote the gradient of $v_{r_{\tilde{q}}}$ at $\mathbf{x}^*$ as $\boldsymbol{\nabla} v_{\tilde{q}}$. If $\boldsymbol{\nabla} v_1, \ldots, \boldsymbol{\nabla} v_q$ are linearly independent, then Assumption 6.2 holds for k.*

Proof. Define subsets of X^l as $\Xi^l_{\tilde{q}} = \{\mathbf{x} \in X^l : v_{r_{\tilde{q}}} > 0\}$, $\tilde{q} = 1, \ldots, q$. Requiring linear independence of $\boldsymbol{\nabla} v_1, \ldots, \boldsymbol{\nabla} v_q$ is a sufficient condition for the existence of 2^q nonempty

subsets of X^l that realize all combinations of $\Xi^l_{\tilde{q}}$ with $\Xi^l_{\hat{q}}$ or $X^l \backslash \Xi^l_{\hat{q}}$, $\hat{q} = 1, \ldots, q$, $\hat{q} \neq \tilde{q}$, for all $l > L$ for some $L \in \mathbb{N}$. Thus, the argument used in the proof of Lemma 6.2 can be extended to show that each possible combination of the bounds on intermediate factors is indeed realized. □

Lemma 6.4. *Consider any k such that $n < k \leq m$ where v_k is defined by $v_k = \psi(v_i)$. Consider a nested sequence of intervals $X^l \to X^* = [\mathbf{x}^*, \mathbf{x}^*]$, $X^l \in \mathbb{ID}$, $X^l \neq X^*$. Suppose either*

1. *that $v_i(\mathbf{x}^*) = 0$ and that for all $l > 0$ there exists a $\mathbf{x}^\dagger_l \in X^l$ and a $\varepsilon_l > 0$ so that $v_i(\mathbf{x}^\dagger_l) = \varepsilon_l$,*
2. *that there exists a $L_1 > 0$ so that $\overline{v}^l_i \leq 0$ for all $l \geq L_1$, or*
3. *that there exists a $L_2 > 0$ so that $\underline{v}^l_i > 0$ for all $l \geq L_2$.*

Then, Assumption 6.2 holds for k.

Proof. Consider Case 1. By assumption, $\underline{v}^l_k = 0$ and $\overline{v}^l_k = 1$, $\forall l$ so that $\lim_{l\to\infty}[\underline{v}^l_k, \overline{v}^l_k] = [0, 1]$. Furthermore, it holds that

$$\begin{aligned}[\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} v_k(\mathbf{x}), \lim_{l\to\infty} \sup_{\mathbf{x}\in X^l} v_k(\mathbf{x})] &= [\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} \psi(v_i(\mathbf{x})), \lim_{l\to\infty} \sup_{\mathbf{x}\in X^l} \psi(v_i(\mathbf{x}))] \\ &= [\psi(v_i(\mathbf{x}^*)), \lim_{l\to\infty} \psi(v_i(\mathbf{x}^\dagger_l))] \\ &= [\psi(0), \lim_{l\to\infty} \psi(\varepsilon_l)] = [0, 1].\end{aligned}$$

Consider Case 2. By assumption, $[\underline{v}^l_k, \overline{v}^l_k] = [0, 0]$ for all $l \geq L_1$. Thus, $v_k(\mathbf{x}) = 0$ for all $\mathbf{x} \in X^{L_1}$ so that $[\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} v_k(\mathbf{x}), \lim_{l\to\infty} \sup_{\mathbf{x}\in X^l} v_k(\mathbf{x})] = [0, 0] = \lim_{l\to\infty}[\underline{v}^l_k, \overline{v}^l_k]$.

Consider Case 3. By assumption, $[\underline{v}^l_k, \overline{v}^l_k] = [1, 1]$ for all $l \geq L_1$. Thus, $v_k(\mathbf{x}) = 1$ for all $\mathbf{x} \in X^{L_2}$ so that $[\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} v_k(\mathbf{x}), \lim_{l\to\infty} \sup_{\mathbf{x}\in X^l} v_k(\mathbf{x})] = [1, 1] = \lim_{l\to\infty}[\underline{v}^l_k, \overline{v}^l_k]$.

Thus, Eq. (6.2) and, hence, Assumption 6.2 hold for factor k. □

Lemma 6.5. *Consider a nested sequence of intervals $X^l \to X^*$, $X^l \in \mathbb{IC}$, $X^l \neq X^*$ and a continuous function $f : X \to \mathbb{R}$. Then,*

$$\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} f(\mathbf{x}) = \inf_{\mathbf{x}\in X^*} f(\mathbf{x}) \qquad \textit{and} \qquad \lim_{l\to\infty} \sup_{\mathbf{x}\in X^l} f(\mathbf{x}) = \sup_{\mathbf{x}\in X^*} f(\mathbf{x}).$$

Proof. Fix $\varepsilon > 0$. Let $\mathbf{x}^*_{\min} \in \arg\min_{\mathbf{x}\in X^*} f(\mathbf{x})$, the infimum is attained since X^* is compact and f is continuous on X^*. Since $X^l \subset X$ is compact and f is continuous on X, f is uniformly continuous on X^l. Uniform continuity of f implies that $\exists \delta > 0$ so that $|f(\mathbf{x}) - f(\mathbf{y})| < \varepsilon$ for all $\mathbf{x}, \mathbf{y} \in X^l$ for which $\|\mathbf{x} - \mathbf{y}\|_2 < \delta\sqrt{n}$. Convergence of X^l to X^* implies that there is a $L > 0$ so that $d_H(X^l, X^*) < \delta$ for all $l > L$. By definition of the Hausdorff metric, $\underline{x}^l_i > \underline{x}^*_i - \delta$ and $\overline{x}^l_i < \overline{x}^*_i + \delta$ for all $l > L$ and $i = 1, \ldots, n$. Thus, $f(\mathbf{x}^\dagger) + \varepsilon > f(\mathbf{x}^\ddagger)$ where $\mathbf{x}^\dagger \in X^l \backslash X^*$ and $\mathbf{x}^\ddagger \in \partial X^*$ with ∂X^* denoting the boundary of X^*. By definition, $f(\mathbf{x}) \geq f(\mathbf{x}^*_{\min})$, $\forall \mathbf{x} \in X^*$ so that $f(\mathbf{x}^\ddagger) \geq f(\mathbf{x}^*_{\min})$. As a result, $f(\mathbf{x}) + \varepsilon > f(\mathbf{x}^*_{\min})$ for all $\mathbf{x} \in X^l$ with $l > L$. Since $X^l \supset X^*$, $\inf_{\mathbf{x}\in X^l} f(\mathbf{x}) \leq f(\mathbf{x}^*_{\min})$ for all

l. ε is arbitrary so that $\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} f(\mathbf{x}) = \inf_{\mathbf{x}\in X^*} f(\mathbf{x})$. An analogous argument can be made to show that $\lim_{l\to\infty} \sup_{\mathbf{x}\in X^l} f(\mathbf{x}) = \sup_{\mathbf{x}\in X^*} f(\mathbf{x})$. □

Lemma 6.6. *Consider any k such that $n < k \leq m$ where v_k is defined by a continuous $o_k \in \mathcal{L}$. Suppose Assumption 6.2 holds for all $i < k$. Consider a nested sequence of intervals $X^l \to X^* = [\mathbf{x}^*, \mathbf{x}^*]$, $X^l \in \mathbb{ID}$, $X^l \neq X^*$. Let $[\underline{v}_i^*, \overline{v}_i^*] = \lim_{l\to\infty} [\underline{v}_i^l, \overline{v}_i^l]$. Then, Assumption 6.2 holds for k if*

$$\min\{o_k(\underline{v}_i^*), o_k(\overline{v}_i^*)\} = \underline{o}_k([\underline{v}_i^*, \overline{v}_i^*]) \quad \textit{and} \quad \max\{o_k(\underline{v}_i^*), o_k(\overline{v}_i^*)\} = \overline{o}_k([\underline{v}_i^*, \overline{v}_i^*]).$$

Proof. First, suppose that v_i is continuous with respect to $\mathbf{x}$ at $\mathbf{x}^*$. Then, $[\underline{v}_i^*, \overline{v}_i^*]$ is a degenerate interval. Since O_k is an interval extension, $O_k([\underline{v}_i^*, \overline{v}_i^*])$ is also a degenerate interval and, hence, Eq. (6.2) holds.

Next, suppose that v_i is not continuous with respect to $\mathbf{x}$ at $\mathbf{x}^*$. Since Assumption 6.2 holds for factor i, $\lim_{l\to\infty} [\underline{v}_i^l, \overline{v}_i^l] = [\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} v_i(\mathbf{x}), \lim_{l\to\infty} \sup_{\mathbf{x}\in X^l} v_i(\mathbf{x})]$ follows. Consider the sequence $V_i^l = [\underline{v}^l, \overline{v}^l]$ converging to $V_i^* = [\underline{v}^*, \overline{v}^*]$. According to Lemma 6.5, it holds that $\lim_{l\to\infty} \inf_{z\in V_i^l} o_k(z) = \inf_{z\in V_i^*} o_k(z)$ and $\lim_{l\to\infty} \sup_{z\in V_i^l} o_k(z) = \sup_{z\in V_i^*} o_k(z)$. The hypothesis of the lemma imply furthermore that $\underline{o}_k([\underline{v}_i^*, \overline{v}_i^*]) = \inf_{z\in[\underline{v}_i^*, \overline{v}_i^*]} o_k(z)$ and that $\overline{o}_k([\underline{v}_i^*, \overline{v}_i^*]) = \sup_{z\in[\underline{v}_i^*, \overline{v}_i^*]} o_k(z)$. Therefore it follows that

$$\begin{aligned}[\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} v_k(\mathbf{x}), \lim_{l\to\infty} \sup_{\mathbf{x}\in X^l} v_k(\mathbf{x})] &= [\lim_{l\to\infty} \inf_{z\in V_i^l} o_k(z), \lim_{l\to\infty} \sup_{z\in V_i^l} o_k(z)] \\ &= [\inf_{z\in V_i^*} o_k(z), \sup_{z\in V_i^*} o_k(z)] = O_k([\underline{v}_i^*, \overline{v}_i^*]) = [\underline{v}_k^*, \overline{v}_k^*],\end{aligned}$$

i.e., Eq. (6.2) holds and, hence, Assumption 6.2 is established for factor k. □

Remark 6.4. An example of a class of univariate functions $u \in \mathcal{L}$ that can meet the hypotheses of Lemma 6.6 are monotone functions. However, the specific implementation of the interval extension $(u, \mathbb{IB}, \mathbb{IR})$ will dictate if u indeed meets the hypotheses of Lemma 6.6.

6.1.6 Relaxations on sequences of intervals

The use of the standard McCormick relaxations in a branch-and-bound algorithm requires further investigation of their behavior with respect to the set on which they are defined. In this section, some properties of the relaxations will be established in such a setting. While the definitions are taken from Scott et al. [156], the facts established hereafter are novel and are not immediate. In the following, it will be assumed that Assumptions 3.1,3.3 and 6.1 hold. As noted earlier, Assumption 6.2 will only be required to show convergence of the bounding operation. It will be pointed out in the statement of the theorem when it is necessary. In the following, a property of the relaxation is first defined and then established by proof. Necessary intermediate results are stated as lemmas.

Definition 6.2. Let $f : D \to \mathbb{R}$ be bounded on $D \in \mathbb{IR}^n$. An algorithm which generates convex and concave relaxations $\check{f}^l$ and $\hat{f}^l$, respectively, of f on any $X^l \in \mathbb{ID}$ is *partition*

monotonic if, for any subintervals $X^{l_2} \subset X^{l_1} \subset C$, $\check{f}^{l_2}(\mathbf{x}) \geq \check{f}^{l_1}(\mathbf{x})$ and $\hat{f}^{l_2}(\mathbf{x}) \leq \hat{f}^{l_1}(\mathbf{x})$, $\forall \mathbf{x} \in X^{l_2}$.

Theorem 6.6. *Standard McCormick relaxations of bounded $\mathcal{L}$-factorable functions are partition monotonic.*

Proof. As shown in [155, Theorem 2.6.5], this follows immediately from inclusion monotonicity of the natural McCormick extension, Theorem 6.3. □

Definition 6.3. An algorithm which generates convex and concave relaxations of $f : D \to \mathbb{R}$ is *weakly partition convergent* if, for any nested and convergent sequence of subintervals of D, $X^l \to X^*$, $X^l \neq X^*$, the sequences convex and concave relaxations of f on X^l, $\{\check{f}^l\}$ and $\{\hat{f}^l\}$, converge uniformly to continuous convex and concave relaxations of f on X^*, $\check{f}^*$ and $\hat{f}^*$, respectively.

Note that this definition deviates from the definition of partition convergent in [156]. Any continuous convex and concave relaxations of f, $\check{f}^*$ and $\hat{f}^*$, meet the definition while Scott et al. [156] require convergence of $\check{f}^l$ and $\hat{f}^l$ to the convex and concave relaxations generated on X^*, respectively.

Lemma 6.7. *Let $\{f^l\}$ be a sequence of functions defined on $D \in \mathbb{IR}^n$ and suppose that $\{f^l\}$ converges pointwise to f on D. If $\{f^l\}$ is nondecreasing, i.e., $f^l(\mathbf{x}) \leq f^{l+1}(\mathbf{x})$, $\forall \mathbf{x} \in X$, and each f^l is lower semi-continuous on X, then f is lower semi-continuous on X.*

Proof. A function $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is lower semi-continuous on $\mathbb{R}^n$ if and only if the level sets $\{\mathbf{x} \in \mathbb{R}^n : g(\mathbf{x}) \leq \gamma\}$ are closed for all $\gamma \in \mathbb{R}$ [82, p. 148]. This allows to extend the proof of Theorem 5.27 in [68] to $X \in \mathbb{IR}^n$ easily. □

Lemma 6.8. *Let $f : D \to \mathbb{R}$ be bounded $\mathcal{L}$-factorable. Suppose $X^l \to X^*$ is a nested sequence of intervals with $X^l \in \mathbb{I}D$, $X^l \neq X^*$ and consider the sequence of convex standard McCormick relaxations of f on X^l, $\{\check{f}^l\}$. Then, $\{\check{f}^l\}$ converges pointwise on X^* to an arbitrary function, denoted as $\check{f}^*$, that is continuous on X^* and a convex relaxation of f on X^*.*

Proof. For any $\mathbf{x} \in X^*$ and $l > 0$, $\check{f}^{l+1}(\mathbf{x}) \geq \check{f}^l(\mathbf{x})$ by Theorem 6.6 and $\check{f}^l(\mathbf{x}) \leq f(\mathbf{x})$ by Theorem 6.3. Thus, $\check{f}^l$ converges pointwise to some function on X^*. This establishes existence of $\check{f}^*$. $\check{f}^l$ is convex by Theorem 6.3. Let $\mathbf{x}, \mathbf{y} \in X^*$ and $\lambda \in [0, 1]$. Set $\mathbf{z} = \lambda\mathbf{x} + (1 - \lambda)\mathbf{y}$. Convexity of $\check{f}^l$ on X^* for all l implies that $\check{f}^*$ is a convex relaxation of f on X^* since

$$\check{f}^*(\mathbf{z}) = \lim_{l\to\infty} \check{f}^l(\mathbf{z}) \geq \lambda \lim_{l\to\infty} \check{f}^l(\mathbf{x}) + (1 - \lambda) \lim_{l\to\infty} \check{f}^l(\mathbf{y}) = \lambda\check{f}^*(\mathbf{x}) + (1 - \lambda)\check{f}^*(\mathbf{y}).$$

As a result of Theorem 6.4, which establishes continuity of $\check{f}^l$, Lemma 6.7 can be applied to find that $\check{f}^*$ is lower semi-continuous on X^*. Lower semi-continuity and convexity of $\check{f}^*$ on X^* imply continuity of $\check{f}^*$ on X^* [143, Theorems 10.2 and 20.5]. □

Theorem 6.7. *Standard McCormick relaxations of bounded $\mathcal{L}$-factorable functions are weakly partition convergent.*

Proof. Suppose X^l is a nested and convergent sequence of subintervals of X, $X^l \to X^*$, and $X^l \neq X^*$. The intervals X^l are closed and bounded by definition and hence compact. Consider the sequence of convex standard McCormick relaxations $\{\check{f}^l\}$. Lemma 6.8 and Theorem 6.6 establish that the relaxations converge pointwise monotonically to a continuous function for each $\mathbf{x} \in X^*$. Rudin [145, Theorem 7.13] shows that this is sufficient for uniform convergence of $\{\check{f}^l\}$ to $\check{f}^*$ on X^*. A similar argument can be made to show $\hat{f}^l \to \hat{f}^*$ uniformly and the theorem follows. □

Definition 6.4. A procedure such as in Definition 6.2 is *degenerate perfect* if $X^* = [\mathbf{x}, \mathbf{x}]$ for any $\mathbf{x} \in D$ implies that $f^\star(\mathbf{x}) = f(\mathbf{x}) = \hat{f}^\star(\mathbf{x})$ where $\check{f}^\star(\mathbf{x})$ and $\hat{f}^\star(\mathbf{x})$ denote the convex and concave relaxations of f on X^*, respectively.

Theorem 6.8. *Standard McCormick relaxations of bounded $\mathcal{L}$-factorable functions are degenerate perfect.*

Proof. This follows directly as natural McCormick extensions are McCormick extensions, Theorem 6.3. □

Remark 6.5. Note that Theorem 6.7 and Theorem 6.8 do not imply that, for any nested sequence of subintervals of X with $\{X^l\} \to [\mathbf{x}^*, \mathbf{x}^*]$, $X^l \neq [\mathbf{x}^*, \mathbf{x}^*]$ and $\mathbf{x} \in X$, $\{\check{f}^l(\mathbf{x}^*)\} \to f(\mathbf{x}^*)$ and $\{\hat{f}^l(\mathbf{x}^*)\} \to f(\mathbf{x}^*)$. While this was asserted in [156], the utilized Lipschitz properties of $\check{f}^l$ and $\hat{f}^l$ do not hold here. Example 6.3 in Section 6.1.3 demonstrates that there are bounded factorable functions where $\{\check{f}^l(\mathbf{x})\} \to \check{f}^*(\mathbf{x}^*) \neq f(\mathbf{x}^*)$.

Theorem 6.9. *Assume f is a bounded $\mathcal{L}$-factorable lower semi-continuous function derived from a bounded $\mathcal{L}$-computational sequence such that Assumption 6.2 holds. Suppose $\{X^l\}$ is a sequence of nested subintervals of D converging to $X^* = [\mathbf{x}^*, \mathbf{x}^*]$, $X^l \neq X^*$. Let $\check{f}^l : X^l \to \mathbb{R}$ be standard McCormick relaxations of $f : D \to \mathbb{R}$ on X^l and let $\mathbf{x}^l_{\min} \in \arg\min_{\mathbf{x} \in X^l} \check{f}^l(\mathbf{x})$. Then, $\lim_{l\to\infty} \check{f}^l(\mathbf{x}^l_{\min}) = f(\mathbf{x}^*)$.*

Proof. Fix $\varepsilon > 0$. Lower semi-continuity of f guarantees that $f(\mathbf{x}^*) \leq \liminf_{\mathbf{x}\to\mathbf{x}^*} f(\mathbf{x})$. Note that $\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l \setminus X^*} f(\mathbf{x}) \geq \liminf_{\mathbf{x}\to\mathbf{x}^*} f(\mathbf{x})$ as, for each l, X^l is a subset of a suitable neighborhood of $\mathbf{x}^*$ referenced in the definition of the lower limit. Therefore, it follows that $f(\mathbf{x}^*) \leq \lim_{l\to\infty} \inf_{\mathbf{x}\in X^l \setminus X^*} f(\mathbf{x})$. Furthermore, it is true that $\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} f(\mathbf{x}) = f(\mathbf{x}^*)$ since $\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} f(\mathbf{x}) = \min\left\{f(\mathbf{x}^*), \lim_{l\to\infty} \inf_{\mathbf{x}\in X^l \setminus X^*} f(\mathbf{x})\right\}$.

Assumption 6.2 implies that $\lim_{l\to\infty} \underline{f}^l = \lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} f(\mathbf{x})$. $\lim_{l\to\infty} \check{f}^l(\mathbf{x}^*)$ exists, see Lemma 6.8, and let it be denoted as $\check{f}^*(\mathbf{x}^*)$. Since $\underline{f}^l \leq \check{f}^l(\mathbf{x}^*)$, it holds that $\lim_{l\to\infty} \underline{f}^l \leq \check{f}^*(\mathbf{x}^*)$. Pointwise convergence of $\check{f}$ implies that there exists $L_1 \in \mathbb{N}$ so that $|\check{f}^l(\mathbf{x}^*) - \check{f}^*(\mathbf{x}^*)| \leq \varepsilon$, $\forall l \geq L_1$. Consequently,

$$f(\mathbf{x}^*) = \lim_{l\to\infty} \inf_{\mathbf{x}\in X^l} f(\mathbf{x}) = \lim_{l\to\infty} \underline{f}^l \leq \check{f}^*(\mathbf{x}^*) \leq \check{f}^l(\mathbf{x}^*) + \varepsilon, \quad \forall l \geq L_1.$$

Continuity of $\check{f}^{L_1}$ guarantees existence of $\delta > 0$ with

$$|\check{f}^{L_1}(\mathbf{x}) - \check{f}^{L_1}(\mathbf{x}^*)| < \varepsilon, \quad \forall \mathbf{x} \in X^{L_1} : \|\mathbf{x} - \mathbf{x}^*\|_2 < \delta.$$

Since $X^l \to X^*$, there exists $L_2 \in \mathbb{N}$ so $\|\mathbf{x} - \mathbf{x}^*\|_2 < \delta$ for all $\mathbf{x} \in X^{L_2}$.

Let $L = \max\{L_1, L_2\}$. Theorem 6.6 and the previous argument imply that

$$\check{f}^{L}(\mathbf{x}) \geq \check{f}^{L_1}(\mathbf{x}) > \check{f}^{L_1}(\mathbf{x}^*) - \varepsilon, \quad \forall \mathbf{x} \in X^{L}.$$

Consequently, $\check{f}^{L_1}(\mathbf{x}^*) - \check{f}^{L}(\mathbf{x}^L_{\min}) \leq \varepsilon$. As a result,

$$f(\mathbf{x}^*) - \check{f}^{L}(\mathbf{x}^L_{\min}) = [f(\mathbf{x}^*) - \check{f}^{L_1}(\mathbf{x}^*)] + [\check{f}^{L_1}(\mathbf{x}^*) - \check{f}^{L}(\mathbf{x}^L_{\min})] \leq 2\varepsilon.$$

Since ε was arbitrary, the theorem follows. □

Remark 6.6. Note that dropping the assumption of lower semi-continuity of f in Theorem 6.9 results in a weaker statement. Since $f(\mathbf{x}^*) \leq \lim_{l\to\infty} \inf_{\mathbf{x}\in X^l\setminus X^*} f(\mathbf{x})$ is not necessarily true then, one can only show that $\check{f}^{l}(\mathbf{x}^l_{\min})$ converges to

$$\min\left\{\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l\setminus X^*} f(\mathbf{x}), f(\mathbf{x}^*)\right\}.$$

If $\mathbf{x}^*$ is in the interior of X^l, $\forall l$, then one can prove convergence to

$$\min\left\{\liminf_{\mathbf{x}\to\mathbf{x}^*} f(\mathbf{x}), f(\mathbf{x}^*)\right\},$$

a statement that does not depend on the sequence of partition elements $\{X^l\}$. In this sense it is more general, but it is also a weaker result since $\lim_{l\to\infty} \inf_{\mathbf{x}\in X^l\setminus X^*} f(\mathbf{x}) \geq \liminf_{\mathbf{x}\to\mathbf{x}^*} f(\mathbf{x})$.

In this section, fundamental properties of the relaxations of bounded factorable functions with discontinuities have been established and assumptions are clarified when these results hold. These results are important when the relaxations are to be used in a branch-and-bound algorithm.

6.2 Branch-and-bound for bounded factorable optimization

In this section, it will be shown that McCormick relaxations of bounded factorable functions can be used to obtain a convergent branch-and-bound algorithm under mild assumptions. Branch-and-bound methods can be used to find a global minimum of a nonconvex nonlinear program. The standard reference for this class of algorithms, Horst and Tuy [88], considers *continuous* functions only when a general theoretical framework is constructed and convergence proofs are established. The work rests on several assumptions for the

bounding, selection and refining operations [88]. The bounding operation is responsible for generating lower and upper bounds on the optimal objective value on a partition element, while the latter two are responsible for selecting a partition element for further investigation and refining it. In the remainder of this section, the discussion will be focused on the bounding operation.

The following definitions are adopted from Horst and Tuy [88] and Horst [87].

Definition 6.5 ([cf. 88, p. 117])**.** Suppose $X \subset \mathbb{R}^n$ and let I be a finite index set. A set $\mathcal{P} = \{X^l : l \in I\}$ with nonempty $X^l \subset X$ is called a *partition* of X with *partition elements* X^l if $X = \bigcup_{l \in I} X^l$ and $X^{l_1} \cap X^{l_2} = \partial X^{l_1} \cap \partial X^{l_2}$, $\forall l_1 \neq l_2 \in I$, where ∂X^{l_1} and ∂X^{l_2} denote the relative boundaries of X^{l_1} and X^{l_2}, respectively.

Definition 6.6. Suppose X^l is an element of a partition $\mathcal{P}$ of X with $X^l \cap E \neq \emptyset$ where E is the feasible set of (1.1). $\alpha(X^l)$ is called an *upper bound* of (1.1) on X^l if $\alpha(X^l) = f(\mathbf{x})$ for some $\mathbf{x} \in X^l \cap E$. Similarly, $\beta(X^l)$ is called a *lower bound* of (1.1) on X^l if $\beta(X^l) \leq \inf_{\mathbf{x} \in X^l \cap E} f(\mathbf{x})$.

Let $k \in \mathbb{N}$ denote the iteration of a branch-and-bound algorithm with partition $\mathcal{P}_k$ of X and corresponding index set I_k. Set $\alpha_0 = +\infty$. Then, $\alpha_k = \min\left\{\alpha_{k-1}, \min_{l \in I_k}\{\alpha(X^l)\}\right\}$ and $\beta_k = \min_{l \in I_k}\{\beta(X^l)\}$ denote the *current upper* and *lower bound* of (1.1) at iteration k, respectively.

Definition 6.7. Suppose $\tilde{X}$ is a partition element of a partition $\mathcal{P}$ of X. A procedure that generates a partition $\mathcal{P}'$ of $\tilde{X}$ with at least two nonempty partition elements is called a *subdivision* or *refinement* of $\tilde{X}$.

Definition 6.8. Suppose $\{\mathcal{P}_k\}$ is an infinite sequence of partitions of X. It is called *successively refined* when, for each k, there exists a $\tilde{X} \in \mathcal{P}_k$ and a $K > k$ so that $\tilde{X}$ has been refined in $\mathcal{P}_K$. A sequence $\{X^k\}$ is called an *infinitely decreasing sequence of successively refined partition elements* when $\{\mathcal{P}_k\}$ is successively refined, $X^k \in \mathcal{P}_k$ and $X^k \supset X^{k+1}$.

Definition 6.9 ([cf. 88, p. 136])**.** The "deletion by infeasibility" rule used in the branch-and-bound algorithm is called *certain in the limit* if for every infinitely decreasing sequence of successively refined partition elements $\{X^l\}$ it holds that $\bar{X} \cap E \neq \emptyset$ where $\bar{X} = \bigcap_l X^l$.

Definition 6.10 ([cf. 88, p. 140])**.** Suppose $\tilde{X}$ is a partition element of a partition $\mathcal{P}$ of X. Consider the sequence of all partition elements $\{\tilde{X}^l\}$, $\tilde{X}^l \subset \tilde{X}$, generated by repeated subdivision of $\tilde{X}$. Let $\delta(X)$ denote the diameter of X, $\delta(X) = \sup_{\mathbf{x},\mathbf{y} \in X} \|\mathbf{x} - \mathbf{y}\|_2$. The subdivision of $\tilde{X}$ is called *exhaustive* if $\lim_{q \to \infty} \delta(\tilde{X}^{l_q}) = 0$ for all nested subsequences $\{\tilde{X}^{l_q}\}$ of $\{\tilde{X}^l\}$, i.e., $\tilde{X} \supset \tilde{X}^{l_1} \supset \tilde{X}^{l_2} \supset \ldots$.

Definition 6.11 ([88, p. 129])**.** Denote as $\mathcal{R}_k$ the set of partition elements of X that have not been fathomed prior to iteration k. If $\inf_{\mathbf{x} \in \tilde{X} \cap E} f(\mathbf{x}) \geq \lim_{k \to \infty} \alpha_k$ for every $\tilde{X} \in \bigcup_{p=1}^{\infty} \bigcap_{k=p}^{\infty} \mathcal{R}_k$, then the selection operation is called *complete*.

Definition 6.12 ([88, p. 130])**.** Suppose $\tilde{X}$ is a partition element of the partition $\mathcal{P}_k$ of X at iteration k so that $\beta(\tilde{X}) = \beta_k$. If $\tilde{X}$ (possibly among other partition elements) is selected for refinement, the selection operation is called *bound improving*.

Definition 6.13 ([88, p. 128])**.** Suppose that, at every iteration k, any unfathomed partition element $\tilde{X} \in \mathcal{P}_k$ can be refined and that any infinitely decreasing sequence of successively refined partition elements $\{X^l\}$ satisfies

$$\lim_{l\to\infty}[\alpha_{k_l} - \beta(X^l)] = 0$$

where k_l denotes the iteration at which partition element X^l is refined. Then, the lower bounding operation is called *consistent*.

Definition 6.14 ([cf. 87, p. 24])**.** Suppose that for any infinitely decreasing sequence of successively refined partition elements $\{X^l\}$ generated by an exhaustive subdivision and satisfying $\lim_{l\to\infty} X^l \to \{\mathbf{x}^\dagger\}$, there exists a subsequence $\{X^{l_q}\}$ such that

$$\lim_{q\to\infty} \beta(X^{l_q}) \geq \min\left\{\liminf_{\mathbf{x}\to\mathbf{x}^\dagger} f(\mathbf{x}), f(\mathbf{x}^\dagger)\right\}.$$

Then, the lower bounding operation is called *strongly consistent*.

In the remainder of the chapter, only partitions using intervals as partition elements are considered. Note that the diameter of an interval X is given by $\delta(X) = \sqrt{\sum_{i=1}^n w(X_i)^2}$. Horst and Tuy point out that bisection of an interval at the midpoint of the longest edge is an exhaustive subdivision [88, p. 144].

6.2.1 Convergence results when minimum is attained

First, it is assumed that f is lower semi-continuous or that f attains its minimum on E.

As remarked by Horst and Tuy, finiteness and convergence properties of the branch-and-bound algorithm depend on the behavior of $\alpha(X^l) - \beta(X^l)$ in the limit [88, p. 128]. Whereas favorable behavior of the McCormick relaxations of factorable functions in this spirit has been argued previously [156], it still needs to be established for the case of bounded factorable functions. In the following, f is assumed to be bounded factorable and the lower bound of (1.1) on a partition element $\tilde{X} \in \mathbb{IC}$, $\beta(\tilde{X})$, is found by constructing the convex McCormick relaxation $\check{f}$ on $\tilde{X}$ and minimizing it, i.e., $\beta(\tilde{X}) = \min_{\mathbf{x}\in\tilde{X}} \check{f}(\mathbf{x})$.

Theorem 6.10 ([cf. 87, p. 28f])**.** *Suppose that Assumption 6.2 holds and that f is lower semi-continuous. Assume that at every step any unfathomed partition element can be refined. Suppose that the subdivision is exhaustive. Then, the lower bounds of* (1.1) *obtained by minimizing the McCormick relaxations are strongly consistent.*

Proof. It is sufficient to show that, for every decreasing sequence of successively refined partition elements $\{X^l\}$ generated by an exhaustive subdivision such that $\lim_{l\to\infty} X^l = \bigcap_l X^l = \bar{X} = [\mathbf{x}^*, \mathbf{x}^*]$, there is a subsequence $\{X^{l_q}\}$ satisfying $\lim_{q\to\infty} \beta(X^{l_q}) = f(\mathbf{x}^*)$. This is guaranteed by Theorem 6.9 since $\beta(X^{l_q}) = \min_{\mathbf{x}\in X^{l_q}} \check{f}^{l_q}(\mathbf{x})$. Lower semi-continuity of f implies that $f(\mathbf{x}^*) \leq \liminf_{\mathbf{x}\to\mathbf{x}^*} f(\mathbf{x})$ and hence strong consistency follows. □

Consider the definition of $\beta(X^l)$ and $\beta_k = \min_{l \in I_k}\{\beta(X^l)\}$ and denote as L_k an element of the index set I_k such that $\beta(X^{L_k}) = \beta_k$. Let $\mathbf{x}_{\min}(X^l) \in \arg\min_{\mathbf{x} \in X^l} \check{f}^l(\mathbf{x})$ and define $\mathbf{x}^k_{\min} \equiv \mathbf{x}_{\min}(X^{L_k})$. Similarly, denote as $\mathbf{x}^k \in E$ a point corresponding to α_k, i.e., $\alpha_k = f(\mathbf{x}^k)$. Horst [87, Theorem 2.1] proves that, for a continuous function f, a strongly consistent lower bounding operation in combination with some additional assumptions is sufficient to show that the lower bound β_k converges to the optimal value of (1.1) and that accumulation points of $\{\mathbf{x}^k_{\min}\}$ solve (1.1). The argument can also be applied to functions that attain their minimum on E.

Theorem 6.11 ([cf. 87, p. 25f])**.** *Suppose that the subdivision of partition elements is exhaustive, that the selection operations is bound improving, that the lower bounding operation is strongly consistent and that the "deletion by infeasibility" rule is certain in the limit. Assume that f attains its minimum on E. Let $X_{\min}$ be the set of accumulation points of $\{\mathbf{x}^k_{\min}\}$. Then, it follows that $\beta = \lim_{k\to\infty} \beta_k = \min_{\mathbf{x}\in E} f(\mathbf{x})$ and $X_{\min} \subset \arg\min_{\mathbf{x}\in E} f(\mathbf{x})$.*

Proof. The proof is identical to the argument in [87] assuming that the minimum is attained is sufficient. Also, the modification of the definition of strongly consistent bounding operations is irrelevant for the proof. □

On the other hand, providing an argument to prove consistency of the lower bounds obtained by using the McCormick relaxations for lower semi-continuous functions is more involved and requires an additional assumption. In the case of a continuous function f, it is obvious that $\alpha(X^l)$ approaches $f(\mathbf{x}^*)$ as $X^l \to \{\mathbf{x}^*\}$ for an infinitely decreasing sequence of successively refined partition elements $\{X^l\}$. When the assumption of continuity of f is dropped, the convergence of $\alpha(X^l)$ to $f(\mathbf{x}^*)$ cannot be asserted as $\alpha(X^l)$ is, by definition, the function value at *some* feasible point in X^l. In particular, it cannot be guaranteed that there exists a $\mathbf{x} \in E$ in a neighborhood of a minimizer of (1.1), denoted as $\mathbf{x}_{\min}$, so that $f(\mathbf{x})$ approximates f^* well. This is demonstrated well in Example 6.1 in Section 6.1.3. For a practical implementation, a subset E' of E in the neighborhood of $\mathbf{x}_{\min}$ must exist so that $f(\mathbf{x})$ is close to f^* when $\mathbf{x} \in E'$. Otherwise it may not be possible to identify numerically a sufficiently good approximation of f^*.

Assumption 6.3. Suppose there exists a $\mathbf{x}_{\min} \in \arg\min_{\mathbf{x}\in E} f(\mathbf{x})$ with the following property: $\mathbf{x}_{\min}$ is not an isolated point of E and for every $\varepsilon > 0$ there is a $\delta > 0$ and a cone E with $\mathbf{x}_{\min}$ at its apex such that $D_\delta = C \cap \{\mathbf{x} \in D : \|\mathbf{x}_{\min} - \mathbf{x}\|_2 < \delta\}$ has nonzero volume and $f(\mathbf{y}) < f^* + \varepsilon$ for all $\mathbf{y} \in D_\delta$.

Remark 6.7. Note that Assumption 6.3 implies that f is upper semi-continuous at $\mathbf{x}_{\min}$ when the domain of f is restricted to a feasible subset of a neighborhood of $\mathbf{x}_{\min}$ with nonzero volume, e.g., a sphere with positive radius in $\mathbb{R}^3$, but not a plane in $\mathbb{R}^3$. However, it does not necessarily imply upper semi-continuity of f at $\mathbf{x}_{\min}$.

Theorem 6.12. *Suppose Assumptions 6.2 and 6.3 hold and that f is lower semi-continuous. Assume that at every step any undeleted partition element can be further refined. Suppose that the subdivision is exhaustive. Then, the lower bounds of* (1.1) *obtained by minimizing the McCormick*

relaxations, i.e., $\beta(\tilde{X}) = \min_{\mathbf{x}\in\tilde{X}} \check{f}(\mathbf{x})$ *for some partition element* $\tilde{X} \in \mathbb{IC}$ *and for the McCormick relaxation* $\check{f}$ *constructed on* $\tilde{X}$*, are consistent.*

Proof. Fix $\varepsilon > 0$. If $\alpha_k < f^* + \varepsilon$ at some iteration k, an ε-optimal solution has been found so that, in combination with Theorem 6.10, consistency of the bounding operation follows.

Otherwise, let $\delta > 0$ and the set C_δ as given by Assumption 6.3. Denote as $\tilde{X}$ the partition element of partition $\mathcal{P}_{k_1}$ with $\tilde{X} \subset C_\delta$ at some iteration k_1. The existence of such a partition element follows from the assumption of exhaustive subdivision and the fact that $\beta(\tilde{X}) < f^* + \varepsilon \leq \alpha_{k_1-1}$ so that neither $\tilde{X}$ nor a partition element that contains $\tilde{X}$, due to Theorem 6.6, could have been fathomed previously. By construction, $\alpha(\tilde{X}) < f^* + \varepsilon$. Thus, a feasible point $\tilde{\mathbf{x}} \in E$ has been found so that $f(\tilde{\mathbf{x}})$ is close to f^*, i.e., $\alpha_{k_1} \leq \alpha(\tilde{X}) < f^* + \varepsilon$ or $\alpha_{k_1} - f^* < \varepsilon$ holds. Consider an infinitely decreasing sequence $\{X^l\}$. Since it is infinitely decreasing, it follows that $\beta(X^l) < \alpha_{k_1}$ for all $l > L_{k_1}$ where L_{k_1} corresponds to iteration k_1; otherwise the partition element would be fathomed hereafter contradicting the assumption that $\{X^l\}$ is an infinitely decreasing sequence. Theorem 6.10 established that $\{\beta(X^l)\}$ converges to f^* so that there exists a k_2 with $f^* - \beta(X^l) < \varepsilon$ for all $l > L_{k_2}$ where L_{k_2} corresponds to iteration k_2. Consequently, $\alpha_{k_l} - \beta(X^l) < 2\varepsilon$ for $l > \max\{L_{k_1}, L_{k_2}\}$. Since ε was arbitrary, the bounding procedure is consistent. □

Horst and Tuy [88] prove the convergence of the sequence of current best points of the branch-and-bound algorithm to an optimal solution. Corollary IV.2 that they present can be extended to lower semi-continuous functions.

Theorem 6.13 ([cf. 88, p. 132])**.** *Let f be lower semi-continuous. Suppose that the bounding operation is consistent and the selection operation is complete. Then every accumulation point of $\{\mathbf{x}^k\}$ solves* (1.1)*.*

Proof. Since E is compact, the sublevel set $C(f(\mathbf{x}^0)) = \{\mathbf{x} \in E : f(\mathbf{x}) \leq f(\mathbf{x}^0)\}$ is bounded and, since f is lower semi-continuous, $C(f(\mathbf{x}^0))$ is closed; cf. [82, p. 148]. Thus, $C(f(\mathbf{x}^0))$ is compact. By construction, $f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k)$, $\forall k$, so that $\{\mathbf{x}^k\} \subset C(f(\mathbf{x}^0))$. Hence, $\{\mathbf{x}^k\}$ possesses accumulation points. The assertion then follows from [88, Theorem IV.2]. □

In this section it was shown that the branch-and-bound algorithm converges under some mild assumptions to a global optimum even in the presence of discontinuities. The presented results assume either that f is lower semi-continuous or that f attains its minimum on E. A discussion of the case when these hypotheses are not met can be found in Appendix 6.2.2.

6.2.2 More general convergence results for branch-and-bound algorithm

In the previous section, it was assumed that f is either lower semi-continuous or attains its minimum on E. Results are outlined below that hold even when these assumptions are generalized.

Remark 6.8. When the assumption that f is lower semi-continuous is dropped in Theorem 6.10, then one cannot appeal to Theorem 6.9. However, with Remark 6.6 in mind, one can argue that

$$\lim_{q\to\infty}\beta(X^{l_q}) = \lim_{q\to\infty}\inf_{\mathbf{x}\in X^{l_q}} f(\mathbf{x}) \geq \min\{\lim_{q\to\infty}\inf_{\mathbf{x}\in X^{l_q}\setminus\{\mathbf{x}^*\}} f(\mathbf{x}), f(\mathbf{x}^*)\} \geq \min\{\liminf_{\mathbf{x}\to\mathbf{x}^*} f(\mathbf{x}), f(\mathbf{x}^*)\},$$

which is sufficient to show that the lower bounding operation is strongly consistent.

Note that Remark 6.8 does not allow for the argument $\lim_{q\to\infty}\beta(X^{l_q}) = \inf_{\mathbf{x}\in X^*\cap E} f(\mathbf{x})$, and consequently $\lim_{k\to\infty}\beta_k = \inf_{\mathbf{x}\in E} f(\mathbf{x})$, when f is not assumed to be lower semi-continuous. In particular, there may be an infinitely decreasing sequence of nested intervals X^l so that there exists a $\mathbf{y}\in\partial E$ with $\mathbf{y}\in\operatorname{int} X^l$, $\forall l$, i.e., all partition elements contain an element of the boundary of the feasible set in its interior. Suppose that $f(\mathbf{y}) = \inf_{\mathbf{x}\in E} f(\mathbf{x})$. Thus, it is conceivable that there exists a $\varepsilon > 0$ and a sequence $\{\mathbf{z}^l\}$ with $\mathbf{z}^l \notin E$, $\mathbf{z}^l \in X^l$, $\forall l$ so that $f(\mathbf{z}^l) < f(\mathbf{y}) - \varepsilon$. As a result, $\lim_{l\to\infty}\beta(X^l) \leq f(\mathbf{y}) - \varepsilon$.

To avoid this complication, another assumption is introduced.

Assumption 6.4. Suppose $f(\mathbf{y}) \geq \inf_{\mathbf{x}\in E} f(\mathbf{x})$, $\forall \mathbf{y}\in X : \mathbf{y}\notin E$.

This assumption can be satisfied by reformulating f as a penalty function, e.g., minimizing $\tilde{f}$ with

$$\tilde{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{if } \mathbf{x}\in E, \\ \overline{f}(E), & \text{otherwise,} \end{cases}$$

where $\overline{f}(E)$ denotes an upper bound, e.g., derived from interval analysis, of f on E.

Remark 6.9. When the assumption that f attains its minimum on D in Theorem 6.11 is removed and Assumption 6.4 holds, one can still argue that $\beta = \inf_{\mathbf{x}\in E} f(\mathbf{x})$ using Theorem 6.9 and Remark 6.6. However, the set of minimizers of f on E, $\arg\min_{\mathbf{x}\in E} f(\mathbf{x})$, is not defined in this case. Instead, consider the set

$$\arg\inf_{\mathbf{x}\in E} f(\mathbf{x}) \equiv \left\{\mathbf{x}\in E : \exists\{\mathbf{z}^l\}\subset E \text{ with } \lim_{l\to\infty}\mathbf{z}^l = \mathbf{x} \text{ and } \lim_{l\to\infty} f(\mathbf{z}^l) = \inf_{\mathbf{z}\in E} f(\mathbf{z})\right\}. \quad (6.3)$$

In this case $X_{\min}\subset \arg\inf_{\mathbf{x}\in E} f(\mathbf{x})$. This can be shown as follows:

Assume that the algorithm does not terminate after a finite number of steps. Consider the sequence of lower bounds $\{\beta_k\}$ with $\mathbf{x}^k_{\min}$, L_k and X^{L_k} as defined previously. From the construction of the algorithm it follows that $\{\beta_k\}$ is a nondecreasing sequence with $\beta_k \leq \inf_{\mathbf{x}\in E} f(\mathbf{x})$. Hence, $\beta = \lim_{k\to\infty}\beta_k$ exists and $\beta \leq \inf_{\mathbf{x}\in E} f(\mathbf{x})$. Let $\mathbf{x}^\dagger_{\min}$ denote an element of the set of accumulation points of the sequence $\{\mathbf{x}^k_{\min}\}$ and let $\{\mathbf{x}^{k_r}_{\min}\}$ be a subsequence of $\{\mathbf{x}^k_{\min}\}$ with subsequential limit $\mathbf{x}^\dagger_{\min}$. Since the partition subdivision is exhaustive and the selection operation is bound improving, a finite number of partition elements is visited in each iteration only. Consequently, a decreasing subsequence of successively refined partition elements $\{X^{q'}\}\subset\{X^{L_{k_r}}\}$ exists such that $\lim_{q'\to\infty} X^{q'} = \{\mathbf{x}^\dagger_{\min}\}$. Since the lower bounding operation is strongly consistent, there exists a subsequence $\{X^q\}\subset\{X^{q'}\}$ such that $\lim_{q\to\infty}\beta(X^q) \geq \min\{\liminf_{\mathbf{x}\to\mathbf{x}^\dagger_{\min}} f(\mathbf{x}), f(\mathbf{x}^\dagger_{\min})\}$. The "deletion

by infeasibility" rule is certain in the limit so that $\mathbf{x}^\dagger_{\min} \in E$. Thus, $\inf_{\mathbf{x}\in E} f(\mathbf{x}) \geq \beta \geq \min\{\liminf_{\mathbf{x}\to\mathbf{x}^\dagger_{\min}} f(\mathbf{x}), f(\mathbf{x}^\dagger_{\min})\}$. By assumption, $f(\mathbf{y}) \geq \inf_{\mathbf{x}\in E} f(\mathbf{x})$ when $\mathbf{y} \notin E$ so that

$$\inf_{\mathbf{x}\in E} f(\mathbf{x}) = \min\{\liminf_{\mathbf{x}\to\mathbf{x}^\dagger_{\min}} f(\mathbf{x}), f(\mathbf{x}^\dagger_{\min})\} = \beta.$$

Thus, the result follows.

Remark 6.10. Assumption 6.3, which implicitly presumes that f attains its minimum on E, is used in Theorem 6.12. The latter can be modified when the minimum of f on E is not attained: define $\tilde{\mathbf{x}}_{\min} \in E$ as the limit of a sequence $\{\mathbf{x}^l\} \subset E$ with $\lim_{l\to\infty} f(\mathbf{x}^l) = f^*$. Suppose that, for every $\varepsilon > 0$, there exists a $\delta > 0$ and a $\mathbf{x} \in E$ for which $\|\mathbf{x} - \tilde{\mathbf{x}}_{\min}\|_2 < \delta$, $\mathbf{x} \neq \tilde{\mathbf{x}}_{\min}$, $f(\mathbf{x}) \leq f^* + \varepsilon$ hold. Under this assumption, consistency of the lower bounding operation can be argued following a proof similar to the one of Theorem 6.12.

Remark 6.11. In Theorem 6.13 it was assumed that f is lower semi-continuous. This assumption was utilized therein to assert that sublevel sets of f are closed. A similar statement is not possible when the assumption of lower semi-continuity of f is dropped as they are equivalent. Consider a discontinuous functions with the following property: there exist two sequences $\{\mathbf{y}^l\}, \{\mathbf{z}^l\} \subset E$ with limits $\mathbf{y}^* \neq \mathbf{z}^*$, respectively, so that $\lim_{l\to\infty} f(\mathbf{y}^l) = f^* = \lim_{l\to\infty} f(\mathbf{z}^l)$ and let $f(\mathbf{y}^*) = f^* \neq f(\mathbf{z}^*)$. The branch-and-bound algorithm is not able to fathom any partition element that contains an infinite number of elements of $\{\mathbf{z}^l\}$. Consequently, $\mathbf{y}^*$ and $\mathbf{z}^*$ are accumulation points of $\{\mathbf{x}^k\}$, whereas, in the strict sense, only $\mathbf{y}^*$ solves (1.1). However, $\mathbf{z}^*$ is in the set $\arg\inf_{\mathbf{x}\in E} f(\mathbf{x})$ as defined by Equation (6.3). Using the argument presented in this remark and asserting Assumption 6.4, one can show that, for any accumulation point $\mathbf{x}^\dagger$ of $\{\mathbf{x}^k\}$, $\mathbf{x}^\dagger \in \arg\inf_{\mathbf{x}\in E} f(\mathbf{x})$ holds.

6.3 Case Studies

In this section, results will be presented from applying the proposed relaxations to some global optimization case studies. First, the discussed method will be applied to a problem from process design and equipment sizing. The section concludes with an example concerning a discrete-time hybrid system.

In the following a simple branch-and-bound algorithm will be used to converge lower and upper bounds and thus find a global optimal solution. At iteration k with partition element $X^l \in \mathcal{P}_k$, upper and lower bounds are found as follows. In general, an upper bound $\alpha(X^l)$ is obtained by evaluating the objective function at the solution of the lower bounding problem (if feasible). To find this solution and a valid lower bound $\beta(X^l)$, different methods are employed. The first method uses only interval arithmetic whereas the other ones use the convex relaxation and a subgradient of the relaxation. The reader is referred to [121] for details on how to construct the subgradient of standard McCormick relaxations.

Method 1 The bound from interval arithmetic, $\underline{f}$, is used as $\beta(X^l)$. The objective function is evaluated at the midpoint of the interval X^l to find $\alpha(X^l)$. This procedure yields

very efficient lower bounds at the expense of tightness.

Method 2 An affine approximation of the convex relaxation of the objective function is constructed sequentially. First, a subgradient of $\check{f}$ is evaluated at the midpoint of X^l and an affine relaxation of f is thus constructed. Combined with the interval bound, $\underline{f}$, CPLEX is used to find a minimum of the affine relaxations. A subgradient of $\check{f}$ is evaluated at this solution, another affine relaxation is added and CPLEX is used to solve this problem. To balance efficiency and accuracy, a total of five minimization problems are solved with CPLEX. The last solution found is reported as $\beta(X^l)$. $\alpha(X^l)$ is obtained by evaluating the objective function at the last point found by CPLEX.

Method 3 Since CPLEX adds considerable overhead, a simple algorithm is explicitly implemented that mimics Method 2 for one-dimensional problems and constructs only two affine relaxations.

Method 4 A bundle solver [113] with bundle size 15 is used to find the minimum of the convex relaxation of the objective function. Note that the QP routines have been modified to prevent an infinite loop in the inner QP. In this case the bundle solver terminates with $\beta(X^l) = -\infty$. $\alpha(X^l)$ is obtained by evaluating the objective function at the point returned by the bundle solver.

In the remainder of this section the different methods will be referred to by these assigned numerals for brevity. The open source C++ library `MC++`, the successor of `libMC` [41, 121], is used to calculate the necessary convex relaxations, and it relies on the interval library `PROFIL` [99] with outward rounding. `MC++` and `PROFIL` are extended to include ψ, its bounds and relaxations as well as subgradients. The global optimization problem is considered converged at iteration k when either $\alpha_k - \beta_k \leq \varepsilon_a$ or $\alpha_k - \beta_k \leq \varepsilon_r |\beta_k|$, where $\varepsilon_a = 10^{-5}$ and $\varepsilon_r = 10^{-5}$ (unless noted otherwise). The best bound heuristic is used to determine the next node and the absolute diameter heuristic is used to select on which variable to branch.

In the case of the more involved problems, the behavior of the proposed methods is compared to the commercial global optimization software BARON [150] as part of GAMS 23.9.5 with regard to number of nodes visited and solution times. Results for the following cases will be presented:

BARON1 Literature model with equal branching priority for each variable.

BARON2 Literature model with branching on binary variables and subset of continuous variables only.

BARON3 Literature model reduced by analytically replacing some equality constraints and intermediate variables; equal branching priority for each variable.

BARON4 Literature model reduced by analytically replacing some equality constraints and intermediate variables; branching on binary variables and subset of continuous variables only.

The same tolerances as listed above are used for BARON. The reader should take note that the branch-and-cut algorithm implemented in BARON employs many features (e.g., range reduction, constraint propagation, etc) that are not implemented in the methods proposed above.

Lastly, a note on notation in this section: in tables containing the results, $\mathbf{x}_{\min}$ always denotes the approximate optimal solution, regardless of symbols used in the problem definition, and f^* indicates the objective value at this point.

6.3.1 Process design and equipment sizing

A specific example from process design in chemical engineering is considered here. Heat exchanger network synthesis problems have been studied extensively, see [65] for a review. A heat exchanger is a device in which two or more fluid streams are brought into energetic contact. Though they cannot exchange mass, the colder stream is heated by the hotter stream and vice versa. The necessary area in the unit for this heat transfer depends on the amount of heat transferred, the temperature difference and the so-called heat transfer coefficient. In the process industry, a common task is to design and size a complex network of heat exchangers to minimize investment and operational cost. Often, heating/cooling utilities such as steam and cooling water are also available. In practice, different device designs are used for different heat transfer areas. As a consequence, the capital cost correlation that links area to cost for these units is not continuous. Also, there are upper limits on the size of a single unit due to the difficulty of transporting large heat exchangers to the plant site. In the present problem, it is assumed that smaller units can be operated in parallel to circumvent this problem.

In the literature, Türkay and Grossmann [168] give a MINLP model that uses disjunctions to model the discontinuity in the cost correlation. An alternative reduced formulation is possible. First, the discontinuous cost correlation can be directly represented without disjunctions or binary variables. Second, equality constraints, in particular energy balances for each heat exchanger, can be used to eliminate variables in the model. Then, one can identify a small number of temperatures that can be chosen independently. After these temperatures are fixed, all remaining intermediate temperatures can be calculated from energy balances. The area A required for each heat exchanger is determined by $A = \frac{Q}{U \times LMTD}$, where $Q = Fc_{p,\mathrm{H}}(T_{\mathrm{H,in}} - T_{\mathrm{H,out}})$ is the heat transfered, Fc_p denotes the heat capacity flow rate, T_{in} and T_{out} the in- and outlet temperatures of the hot and cold streams, U the overall heat transfer coefficient, and $LMTD$ the log mean temperature difference. Instead of using the exact expression for $LMTD$, Chen's approximation [42] is used,

$$LMTD = \left((T_{\mathrm{out},H} - T_{\mathrm{in},C})(T_{\mathrm{in},H} - T_{\mathrm{out},C}) \frac{(T_{\mathrm{out},H} - T_{\mathrm{in},C}) + (T_{\mathrm{in},H} - T_{\mathrm{out},C})}{2} \right)^{\frac{1}{3}}.$$

Depending on the heat transfer area, one can then choose from three different available heat exchanger designs with different investment cost correlations, which are given in Table 6.2. When the necessary area for one heat exchanger exceeds the maximum area of

A [m^2]	investment cost [\$/yr]
$0 \leq A \leq 20$	$670A^{0.83}$+2000
$20 < A \leq 50$	$640A^{0.83}$+8000
$50 < A \leq 100$	$600A^{0.83}$+16000

Table 6.2: Equipment cost correlation for heat exchangers depending on required area

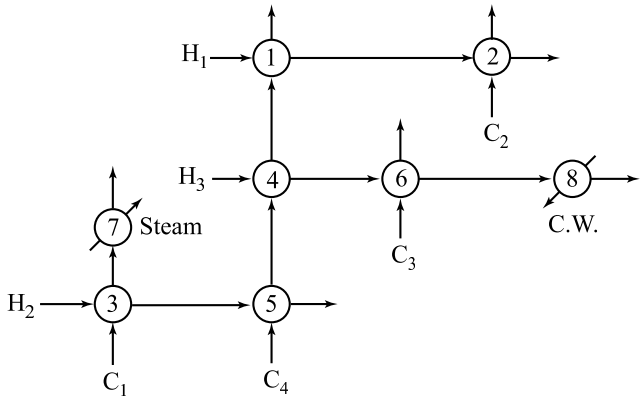


Figure 6.5: Structure of heat exchanger network 1

the largest heat exchanger design, the streams will be split and several heat exchangers will be used. At most seven parallel heat exchangers will be allowed to always ensure feasibility of the solution. Lastly, the operating expenses are found by calculating the cost of cooling water (20 \$/kW yr) and the cost of steam (80 \$/kW yr).

Overall, a factorable representation of the objective function can be constructed as outlined above. In the routine to calculate the convex relaxations of the objective function, a priori known bounds on intermediate quantities, e.g., areas need to be nonnegative, temperature differences in the heat exchangers cannot be negative and intermediate temperatures must be between inlet and outlet temperatures of the respective stream, are used to obtain tighter bounds for intermediate expressions.

Heat exchanger network 1

The first case study was taken from [168]. Consider the heat exchanger network depicted in Figure 6.5 with stream data given in Table 6.3. Let the overall heat transfer coefficient of the heat exchangers be given by $(U_i) = (1.5, 0.2, 0.06, 1.6, 0.04, 0.3, 0.6, 1.7)$ kW/m^{2}K.

There are seven unknown intermediate stream temperatures and two unknown utility heat loads. Since one can write an energy balance for each of the eight heat exchangers, the problem has one degree of freedom. The temperature of stream H_3 at the outlet of exchanger 6 was selected as the decision variable T. From requirements for feasible heat exchange, i.e., no temperature crossover in the heat exchangers, it follows that $T \in [382.25, 499.36]$ K. Once this variable is fixed, the remaining intermediate temperatures,

Stream	Fc_p [kW/K]	T_{in} [K]	T_{out} [K]
H_1	30.0	626	586
H_2	13.5	620	519
H_3	20.0	528	353
C_1	31.0	525	613
C_2	5.0	405	576
C_3	28.0	353	386
C_4	11.0	313	545
steam	—	650	650
cooling water	—	293	308

Table 6.3: Data for process and utility streams in heat exchanger network 1

Method	# LBPs	# UBPs	Runtime [s]	f^*	$\mathbf{x}_{\text{min}}$ [K]
1	399	205	0.2269	411,809	418.103
2	63	42	0.2776	411,809	418.104
3	67	41	0.076	411,809	418.103
4	61	40	0.3334	411,809	418.104
BARON1	48 iterations		3.96	411,809	418.103
BARON2	91 iterations		4.92	411,809	418.103
BARON3	46 iterations		1.97	411,809	418.103
BARON4	18 iterations		1.22	411,809	418.103

Table 6.4: Comparison of different methods with BARON for the first heat exchanger case study

utility heat loads, areas and hence investment costs can be computed by a factorable function as described before.

The solutions as found with the different methods are compared in Table 6.4 to the solution obtained with BARON [150] using the MINLP model proposed in [168]. In the case of the reduced model, the energy balances are solved for the intermediate temperatures, which are subsequently substituted in the equation for $LMTD$. The expressions for $LMTD$ have not been substituted since the non-integer exponent is reformulated by GAMS using the exponential function and the natural logarithm. During the model development, GAMS aborted reporting domain violations so that this substitution is not feasible. In the case of selective branching, BARON is instructed to branch on T and the binary variables only.

All methods find the same solution; see Table 6.4 for more details. Lastly, it is instructive to point out that the full disjunctive model introduces 168 binary and 360 continuous variables.

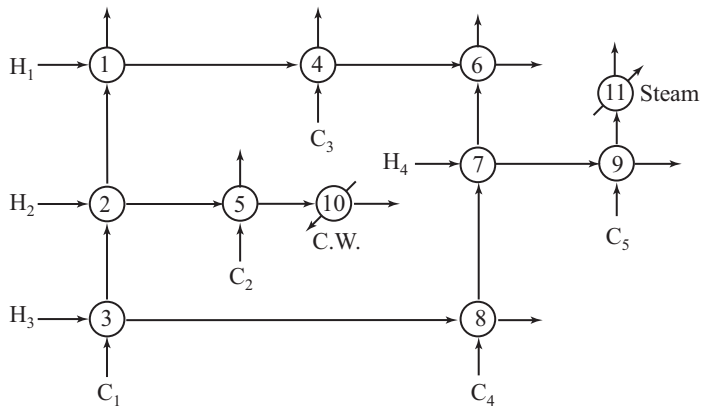


Figure 6.6: Structure of heat exchanger network 2

Stream	Fc_p [kW/K]	T_{in} [C]	T_{out} [C]
H_1	22.4	400	150
H_2	12.0	180	40
H_3	26.0	150	45
H_4	24.0	135	100
C_1	15.0	105	360
C_2	20.0	40	65
C_3	22.0	90	190
C_4	35.0	25	110
C_5	16.2	30	150
steam	—	400	400
cooling water	—	15	30

Table 6.5: Data for process and utility streams in heat exchanger network 2

Heat exchanger network 2

Consider the heat exchanger network depicted in Figure 6.6 with stream data given in Table 6.5. The goal is to optimize the network and size the equipment so that the combined investment and operational cost is minimized. Let the overall heat transfer coefficient be given by $(U_i) = (1.0, 0.1, 2.1, 0.05, 1.0, 0.2, 1.5, 0.7, 4.0, 1.2, 0.1)\,\text{kW/m}^2\text{K}$. There are eleven unknown intermediate stream temperatures and two unknown utility heat loads. Since one can write an energy balance for each of the eleven heat exchangers, the problem has two degrees of freedom. The temperature of stream H_3 at the outlet of exchanger 3 and the temperature of stream H_2 at the outlet of exchanger 2 were selected as the decision variables T' and T'', respectively. From requirements for feasible heat exchange, i.e., no temperature crossover in the heat exchangers, it follows that $T' \in [129.81, 150.0]\,\text{C}$ and

Method	# LBPs	# UBPs	Runtime [s]	f^*	$\mathbf{x}_{\min}$ [C]
1	$> 100,000$	$> 70,836$	> 92.0	—	—
2	3,290	1,804	21.6	599,740	(130.40, 160.49)
4	3,661	2,000	25.9	599,740	(130.40, 160.49)
BARON1	65 iterations		8.55	599,740	(130.40, 160.49)
BARON2	151 iterations		13.80	599,740	(130.40, 160.49)
BARON3	389 iterations		33.18	599,740	(130.40, 160.49)
BARON4	218 iterations		23.41	599,740	(130.40, 160.49)

Table 6.6: Comparison of different methods with BARON for the second heat exchanger case study

$T'' \in [124.17, 180.0]$ C; furthermore, it needs to hold that

$$26T' + 15T'' \geq 5625$$
$$312T' + 210T'' \leq 84565.$$

The solutions as found with the different methods are compared in Table 6.6 to the solution obtained with BARON [150] using the model with disjunctions proposed in [168]. The reduced model is constructed as outlined in Section 6.3.1. In the case of selective branching, BARON is instructed to branch on T', T'' and the binary variables only.

A few remarks are in order. First, the interval bounds do not converge to the solution in 100,000 iterations and consequently, the branch and bound procedure in Method 1 fails to terminate with a guaranteed solution. Second, note that BARON requires fewer iterations than Methods 2 and 4 to identify its solution, however, each iteration is significantly more costly; see Table 6.6 for more details. Also note that the full disjunctive model used in BARON introduces 231 binary and 496 continuous variables.

6.3.2 Discrete-time hybrid systems

A second class of problems with discontinuous behavior is considered. Hybrid systems combine continuous dynamics, that are described by differential equations, and discrete dynamics, which are discontinuous changes in state variables or switching of the dynamic model triggered by so-called events [49, 67]. In discrete-time systems, the continuous dynamics are discretized and described by difference equations. The problem below, which concerns the optimal control of a linear discrete-time hybrid system, is slightly adapted from [112]. Consider the global optimization problem with an embedded discrete-time

Case	Method	# LBPs	# UBPs	Runtime [s]	f^*
1	1	$> 100,000$	$> 100,000$	> 30.0	—
	2	1	1	0.0280	7.256
	4	1	1	0.0114	7.261
2	1	$> 100,000$	$> 69,712$	> 28.2	—
	2	1	1	0.0263	13.077
	4	1	1	0.0112	13.049

Table 6.7: Comparison of different methods for both cases of the discrete-time hybrid system. Note that Method 1 does not converge in either case after solving 100,000 iterations.

hybrid system

$$\begin{aligned} \min_{u_0,\ldots,u_{N-1}} \quad & \sum_{k=1}^{N} \left(\mathbf{x}_k^{\mathrm{T}} \mathbf{R} \mathbf{x}_k + u_{k-1} Q u_{k-1} \right) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{A}(m(k))\mathbf{x}_k + \mathbf{B}(m(k))u_k,\ k = 1,\ldots,N-1, \\ & m(k) = \begin{cases} 1 & \text{if } x_{k,1} \leq x_{k,2}, \\ 2 & \text{otherwise,} \end{cases},\ k = 1,\ldots,N-1, \end{aligned}$$

with $N = 10$ and, for $k = 1,\ldots,N$, $u_{k-1} \in [-1,1]$,

$$\mathbf{A}(m(k)) = \begin{cases} \begin{bmatrix} 0 & 0.2 \\ -0.4 & -0.06 \end{bmatrix} & \text{if } m(k) = 1, \\ \begin{bmatrix} 0.2 & 0.6 \\ -0.2 & 0.4 \end{bmatrix} & \text{if } m(k) = 2, \end{cases} \qquad \mathbf{B}(m(k)) = \begin{cases} \begin{bmatrix} 0 \\ 0.4 \end{bmatrix} & \text{if } m(k) = 1, \\ \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix} & \text{if } m(k) = 2, \end{cases}$$

$$\mathbf{R} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.5 \end{bmatrix}, \qquad \text{and } Q = 1.0.$$

Two cases are considered where the initial conditions differ: Case 1 with $\mathbf{x}_0 = [4,5]^{\mathrm{T}}$ and Case 2 with $\mathbf{x}_0 = [5,4]^{\mathrm{T}}$.

The objective can be calculated using a finite algorithm that takes $u_0,\ldots,u_{N-1}$ as input and returns the objective. The detailed results for both cases are shown in Table 6.7. Here, the relative tolerance is set to $\varepsilon_r = 10^{-1}$ initially. Note that in both cases, Methods 2 and 4 find the optimal solution at the root node while Method 1 does not converge the lower and upper bound within 100,000 iterations, which is indicative of the weakness of only using interval methods for multi-dimensional problems.

It is important to remark that, although both problems are solved at the root node, the lower bound does not converge to the value of the optimal solution. Instead, a small but finite discrepancy will remain indefinitely. This results from the presence of discontinuities; cf. the discussion in Section 6.1.4. In this example, it does not impact convergence when $\varepsilon_r = 10^{-1}$. However, Methods 2 and 4 do not converge Case 1 within 100,000 iterations

when $\varepsilon_r = 10^{-2}$ and Case 2 when $\varepsilon_r = 10^{-4}$.

6.4 Conclusion

A procedure to construct interval bounds and convex and concave relaxations of factorable functions with discontinuities has been presented. McCormick's composition theorem [118] is extended to bounded, but not necessarily continuous, functions. The crux of the proposed extension lies in the observation that discontinuities can be modeled using a step function [182] and that convex and concave envelopes can be readily constructed for this function. Furthermore, it was shown that most theoretical results developed for the continuous case [156] hold even when the assumption of continuity is dropped. Only establishing convergence of a sequence of relaxations to the function when a sequence of intervals converging to a degenerate interval is considered requires additional assumptions. Currently, some results are established to show when this assumption holds. Nevertheless, this remains an active area of research for the authors as, e.g., examples shown in the previous section indicate that the relaxations converge for problems of practical importance or at least provide sufficiently tight relaxations. Also, these case studies show that the proposed method may provide a very effective means to solve optimization problems with discontinuities to global optimality without introducing binary variables. Thus an increase in size of the global optimization problem can be avoided, which is very desirable since known global optimization algorithms scale exponentially. Also note that so far no range reduction techniques have been employed which considerably improve convergence in BARON. There appears no distinct advantage of reducing the size of the optimization problem in BARON. Also, it is not possible to deduce a general advantage when branching on a subset of the continuous variables only. Lastly, the advantage of convex relaxations over interval bounds is demonstrated for multi-dimensional problems. While one-dimensional problems can be solved efficiently when only interval bounds are available, the convex relaxations are key for efficiently finding the optimal solutions of multi-dimensional problems.

Chapter 7

Improving convergence of relaxations of bounded $\mathcal{L}$-factorable functions

In Chapter 6, a method to obtain relaxations for a class of discontinuous functions was introduced. While incorporating a larger class of functions into the notion of a factorable function allows formulation of more problems with a reduced number of variables visible to the optimizer, cf. the idea of global optimization of algorithms introduced by Mitsos et al. [121], it also introduces new challenges. For example, once the continuity assumption on the functions present in (1.1) is dropped, standard results, e.g., in interval analysis, are not necessarily true anymore. It is possible to construct Lipschitz interval extensions of most *continuous* univariate functions and then use composition results to combine these to obtain Lipschitz interval extensions of complex functions [122]. The Lipschitz property guarantees that the interval extensions converge to the underlying real-valued function as the host set converges to a degenerate interval. Certainly, a discontinuous function does not possess a Lipschitz constant so that convergence of the interval arithmetic cannot be established using this route.

Recall Example 6.3 which exemplified that convergence of the proposed relaxations can not be taken for granted. One source of this behavior is the dependency problem in interval analysis. Since rigorous bounds are calculated as a zeroth order approximation, all information about interdependence of factors is lost. Thus, the worst-case must always be accounted for, leading to overestimation. In the case of Lipschitz continuous functions, the overestimation of the natural interval extensions can be shown to decrease linearly as the host sets shrinks, cf. Theorem 3.2. This is no longer necessarily true for discontinous functions. In Section 6.1.5 it was discussed under which conditions convergence of the relaxations can be nevertheless guaranteed. However, these conditions are not easy to verify for more complex problems.

In this chapter, a solution to overcome this behavior will be introduced in Section 7.1. In Section 7.3, results for two case studies will be reported. The chapter closes with discussion of the results in Section 7.4 and conclusions in Section 7.5.

7.1 Branching on discontinuous factors

In this section, a solution to this problem will be proposed to allow the use of the relaxations of discontinuous factorable functions in global optimization algorithms. It will

be argued that one can branch on an intermediate discontinuous factor. Factors with the discontinuous univariate function ψ will be fixed at either 0 or 1 and both possibilities will be added as new nodes to the branch-and-bound stack. This operation can be interpreted as branching on binary variables during the solution of an equivalent mixed-integer program.

This section will use many technical terms from the realm of branch-and-bound theory, see 6.2 for precise definitions.

In this chapter, unless noted otherwise, lower bounds are obtained by using McCormick relaxations [121, 156], i.e.,

$$\beta(X_k) = \inf\{\check{f}(\mathbf{x}) : \mathbf{x} \in X_k, \check{\mathbf{g}}(\mathbf{x}) \leq \mathbf{0}\}$$

where $\check{f}$ and $\check{\mathbf{g}}$ are the convex standard McCormick relaxations of f and $\mathbf{g}$ on X_{k_q}.

In the remainder of this section, it will be first shown that this new branching operation continues to generate valid lower bounds. Then, it will be argued that this scheme is applicable in a branch-and-bound procedure without tampering with convergence. Some details regarding its implementation are given at the end of this section.

7.1.1 Validity of obtained lower bounds

Theorem 7.1. *Suppose $D \subset \mathbb{R}^n$ is convex and let $h : D \to \mathbb{R}$ be a bounded $\mathcal{L}$-factorable function. Suppose $(\mathcal{S}, \psi_0)$ is a corresponding bounded $\mathcal{L}$-computational sequence with factors $\{(v_k, D_k, \mathbb{R})\}_{k=1}^{n_h}$. Assume that $\exists \hat{k}$, $n < \hat{k} \leq n_h$ so that $o_{\hat{k}} = \psi$. Consider $h^1 : X \to \mathbb{R}$ and $h^0 : X \to \mathbb{R}$ which have the identical factorable representation as h except that $v_{\hat{k}} = 1$ in h^1 and $v_{\hat{k}} = 0$ in h^0. Let $\check{h}^1$ and $\check{h}^0$ be convex relaxations of h^1 and h^0 on $X \in \mathbb{I}D$. Define $h^\dagger = \min_{\mathbf{x}\in X} \check{h}^1(\mathbf{x})$ and $h^\ddagger = \min_{\mathbf{x}\in X} \check{h}^0(\mathbf{x})$. Then,*

$$\min\{h^\dagger, h^\ddagger\} \leq \inf_{\mathbf{x}\in X} h(\mathbf{x}).$$

Proof. Surely, it holds that

$$h(\mathbf{x}) = \begin{cases} h^1(\mathbf{x}) & \text{if } \pi_{\hat{k}}(v_1(\mathbf{x}), \ldots, v_{\hat{k}-1}) > 0, \\ h^0(\mathbf{x}) & \text{otherwise.} \end{cases}$$

Since $\check{h}^1$ and $\check{h}^0$ are convex relaxations of h^1 and h^0 on X, respectively, and by definition of $h^\dagger$ and $h^\ddagger$, it holds for all $\mathbf{x} \in X$ that

$$h^\dagger \leq \check{h}^1(\mathbf{x}) \leq h^1(\mathbf{x}) \text{ and}$$
$$h^\ddagger \leq \check{h}^0(\mathbf{x}) \leq h^0(\mathbf{x}).$$

This implies for all $\mathbf{x} \in X$ that

$$\min\{h^\dagger, h^\ddagger\} \leq h^1(\mathbf{x}) \text{ and}$$
$$\min\{h^\dagger, h^\ddagger\} \leq h^0(\mathbf{x}).$$

Thus, $\min\{h^\dagger, h^\ddagger\} \leq h(\mathbf{x})$ for all $\mathbf{x} \in X$ so that the assertion follows. □

An immediate consequence of Theorem 7.1 is the following:

Corollary 7.1. Suppose there is an unfathomed node identified by its host set $\tilde{X}$ in the branch-and-bound tree used to solve (1.1). One discontinuous univariate factor of f or $\mathbf{g}$ has been replaced as described in Theorem 7.1 to obtain f^1, f^0 and $\mathbf{g}^1$, $\mathbf{g}^0$. $\tilde{X}$ can be replaced by two nodes, identified as $\tilde{X}^1$ and $\tilde{X}^0$, with $\tilde{X}^1 = \tilde{X}$ and $\tilde{X}^0 = \tilde{X}$ on which f and $\mathbf{g}$ are replaced by f^1 and $\mathbf{g}^1$ and by f^0 and $\mathbf{g}^0$, respectively, in the lower bounding problems. This manipulation is valid in the sense that

$$\beta(\tilde{X}) \geq \min\{\beta(\tilde{X}^1), \beta(\tilde{X}^0)\}.$$

Proof. This follows immediately from Theorem 7.1. □

This manipulation of the nodes in the branch-and-bound tree will be called *branching on a discontinuous factor*, though the host sets of the child nodes are identical to the one of the parent node. Also, note that the functions f and $\mathbf{g}$ in the upper bounding problem will remain unchanged.

Remark 7.1. It is important to point out that it is not necessary to branch on discontinuous factors that are already uniquely determined on $X \in \mathbb{I}D$, i.e., ones for which $\underline{w}_{\hat{k}}(X) > 0$ or $\overline{w}_{\hat{k}}(X) \leq 0$ where $W_{\hat{k}}(X) = \pi_{\hat{k}} \circ (V_1(X), \ldots, V_{\hat{k}-1}(X))$.

Remark 7.2. It follows by induction that the procedure outlined in Theorem 7.1 can be repeated until all discontinuous factors have been branched on or are uniquely determined in the lower bounding problem. More formally, let I be the index set that identifies each discontinuous univariate factor of f and $\mathbf{g}$, i.e., $o_i = \psi$, $i \in I$ where o_i refers to factors of f or $\mathbf{g}$. Denote as $f_\varsigma : D \to \mathbb{R}$ and $\mathbf{g}_\varsigma : D \to \mathbb{R}$ the continuous factorable functions derived from f and $\mathbf{g}$ by branching on discontinuous univariate factors. Furthermore, introduce ς, a vector that identifies if and which branch of f or $\mathbf{g}$ are chosen in f_ς or $\mathbf{g}_\varsigma$. It is defined as follows: $\varsigma_i = -1$ if no branching occurred, $\varsigma_i = 1$ if $v_i = 1$ and $\varsigma_i = 0$ if $v_i = 0$ for each $i \in I$. Lastly, let $X \in \mathbb{I}D$ and define

$$Y_\varsigma(X) = \{\mathbf{x} \in X : \underline{w}_i(X) > 0 \text{ if } \varsigma_i = 1, \forall i \in I \text{ and } \overline{w}_i(X) \leq 0 \text{ if } \varsigma_i = 0, \forall i \in I\},$$

the set of points at which ς chooses the correct branch of f and $\mathbf{g}$ in the limit of a degenerate interval X.

Note that the factors refer to either the factorable representations of f or $\mathbf{g}$. It is tacitly assumed that the correct identification of the corresponding function is obvious.

Definition 7.1. Let I and ς be as defined in Remark 7.2. Suppose $\mathbf{x}^* \in D$. It will be said that *the branch chosen by ς coincides with the evaluation of f and $\mathbf{g}$ at $\mathbf{x}^*$* if $v_i(\mathbf{x}^*) = \varsigma_i$, $\forall i \in I$.

7.1.2 Consistency of bounding operation

Recall the definition of a consistent bounding operation (see Definition 6.13) that is useful to establish convergence of the branch-and-bound algorithm, see Theorem 6.12 and [88, Theorem IV.3]. One route to establish consistency of the bounding operation is to first establish that it is strongly consist (see Definition 6.14). It is also important to establish that the deletion by infeasibility rule used in the branch-and-bound algorithm is certain in the limit (see Definition 6.9). In particular, note that this rule also applies to nodes that are infeasible because the discontinuous factors are fixed to the incorrect branch of f or $\mathbf{g}$. Recall that E refers to the feasible set of (1.1).

Lemma 7.1. *Suppose that $\{X_l\} \subset D$ is a nested sequence of intervals with $\lim_{l\to\infty} X_l = [\mathbf{x}^*, \mathbf{x}^*]$ so that $\mathbf{x}^* \in E$. Assume that f is lower semi-continuous at $\mathbf{x}^*$. Then,*

$$f(\mathbf{x}^*) = \lim_{l\to\infty} \inf\{f(\mathbf{x}) : \mathbf{x} \in X_l, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}. \tag{7.1}$$

Proof. First, define the sets $E_l \equiv \{\mathbf{x} \in X_l : \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$. Since $\mathbf{x}^* \in E \cap X_l$, $\forall l$, it follows that $D_l \neq \emptyset$, $\forall l$. In particular, this implies that

$$f(\mathbf{x}^*) \geq \inf_{\mathbf{x}\in E_l} f(\mathbf{x}),\ \forall l. \tag{7.2}$$

Since $X_l \supset X_{l+1}$, it follows that $E_l \supset E_{l+1}$ so that $\inf_{\mathbf{x}\in E_l} f(\mathbf{x}) \leq \inf_{\mathbf{x}\in E_{l+1}} f(\mathbf{x})$. In combination with Eq. (7.2), this establishes the existence of the limit in Eq. (7.1).

Next, let $\delta > 0$ be arbitrary and consider the ball $N_\delta(\mathbf{x}^*) \equiv \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{x}^*\|_2 \leq \delta\}$. Since E_l does not necessarily contain all points in a neighborhood of $\mathbf{x}^*$, $N_\delta(\mathbf{x}^*) \cap E_l \subset N_\delta(\mathbf{x}^*)$. It follows that

$$\min\{f(\mathbf{x}^*), \liminf_{\mathbf{x}\to\mathbf{x}^*} f(\mathbf{x})\} \leq \liminf_{l\to\infty} \inf_{\mathbf{x}\in E_l} f(\mathbf{x}). \tag{7.3}$$

Lastly, note that lower semi-continuity of f at $\mathbf{x}^*$ implies by definition that

$$f(\mathbf{x}^*) \leq \liminf_{\mathbf{x}\to\mathbf{x}^*} f(\mathbf{x}). \tag{7.4}$$

Thus, combining Eqs. (7.2), (7.3) and (7.4) yields Eq. (7.1), the desired result. □

Lemma 7.2. *Suppose that $\{X_l\}$ is a nested sequence of intervals with $\lim_{l\to\infty} X_l = [\mathbf{x}^*, \mathbf{x}^*]$ so that $\mathbf{x}^* \in E$. Let I, ς and $Y_\varsigma(\cdot)$ as defined in Remark 7.2. Suppose that $f_\varsigma : X \to \mathbb{R}$ and $\mathbf{g}_\varsigma : X \to \mathbb{R}$ are continuous factorable functions derived from f and $\mathbf{g}$ by branching on all discontinuous univariate factors and assume that all remaining univariate functions are Lipschitz. Assume that ς coincides with the evaluation of f and $\mathbf{g}$ at $\mathbf{x}^*$. Let $\check{f}_\varsigma^l$ and $\check{\mathbf{g}}_\varsigma^l$ denote the convex standard McCormick*

relaxations of f_ς and $\mathbf{g}_\varsigma$ on X_l. Then,

$$f(\mathbf{x}^*) = \lim_{l\to\infty} \inf\{\check{f}_\varsigma^l(\mathbf{x}) : \mathbf{x} \in Y_\varsigma(X_l),\ \check{\mathbf{g}}_\varsigma^l(\mathbf{x}) \leq \mathbf{0}\}. \tag{7.5}$$

Proof. As X_l converges to the degenerate interval $[\mathbf{x}^*, \mathbf{x}^*]$ and since f_ς and $\mathbf{g}_\varsigma$ are Lipschitz continuous, the convex relaxations $\check{f}_\varsigma^l$ and $\check{\mathbf{g}}_\varsigma^l$ converge to f_ς and $\mathbf{g}_\varsigma$ at $\mathbf{x}^*$ [156, Theorem 5]. Thus,

$$\lim_{l\to\infty} \check{f}_\varsigma^l(\mathbf{x}^*) = f_\varsigma(\mathbf{x}^*). \tag{7.6}$$

Define the sets $E_l' \equiv \{\mathbf{x} \in Y_\varsigma(X_l) : \check{\mathbf{g}}_\varsigma^l(\mathbf{x}) \leq \mathbf{0}\}$. By assumption, $\lim_{l\to X_l} = [\mathbf{x}^*, \mathbf{x}^*]$, $\mathbf{x}^* \in E$ and the fact that ς coincides with the evaluation of f and $\mathbf{g}$ at $\mathbf{x}^*$ imply that $\lim_{l\to\infty} E_l' = \{\mathbf{x}^*\}$ which yields together with Eq. (7.6) that

$$\lim_{l\to\infty} \inf_{\mathbf{x}\in E_l'} \check{f}_\varsigma^l(\mathbf{x}) = \lim_{l\to\infty} \check{f}_\varsigma^l(\mathbf{x}^*) = f_\varsigma(\mathbf{x}^*).$$

Lastly, $f_\varsigma(\mathbf{x}^*) = f(\mathbf{x}^*)$ so that Eq. (7.5) follows. □

Remark 7.3. Suppose that there exists j so that $g_j(\mathbf{x}^*) > 0$. Hence, $\mathbf{x}^* \notin E$. Lower semi-continuity of g_j at $\mathbf{x}^*$ implies that there exists a $\delta > 0$ so that $g_j(\mathbf{x}) > 0$ for all $\mathbf{x}$ with $\|\mathbf{x} - \mathbf{x}^*\|_2 \leq \delta$. Hence, $f(\mathbf{x}^*) < \lim_{l\to\infty} \inf\{f(\mathbf{x}) : \mathbf{x} \in X_l, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\} = +\infty$.

Lemma 7.3. *Assume that the partitioning is exhaustive and that the deletion by infeasibility rule is certain in the limit. Suppose ς has been fixed, $\varsigma_i = \{0, 1\}$, $i \in I$ and consider an infinitely decreasing sequence of successively refined nodes, $\{X_{k_q}\}$. Then, the branch chosen by ς coincides with the evaluation of f and $\mathbf{g}$ at $\mathbf{x}^*$, $[\mathbf{x}^*, \mathbf{x}^*] = \bigcap_{q=1}^\infty X_{k_q}$.*

Proof. Exhaustive partitioning of the nodes implies that $\bar{X} = \bigcap_{q=1}^\infty X_{k_q} = [\mathbf{x}^*, \mathbf{x}^*]$ for some $\mathbf{x}^* \in X$. Note that $\bar{X}$ is a degenerate interval and that interval extensions are exact on degenerate intervals. Therefore, $\underline{v}_i(\bar{X}) = v_i(\mathbf{x}^*) = \overline{v}_i(\bar{X})$. Suppose that ς does not coincide with the evaluation of f and $\mathbf{g}$ at $\mathbf{x}^*$. In particular, let there exist an $i \in I$ with $v_i(\mathbf{x}^*) \neq \varsigma_i$. Without loss of generality assume that $\varsigma_i = 1$ and hence $v_i(\mathbf{x}^*) = 0$. It follows that $w_i(\mathbf{x}^*) \leq 0$ and thus $\overline{w}_i(\bar{X}) \leq 0$. Thus, $\mathbf{x}^* \notin Y_\varsigma(\bar{X})$. Since $\bar{X} = \{\mathbf{x}^*\}$ and $Y_\varsigma(\bar{X}) \subset \bar{X}$, it follows that $Y_\varsigma(\bar{X}) = \emptyset$. This contradicts the assumption that the deletion by infeasibility rule is certain in the limit. □

Theorem 7.2. *Suppose that f is lower semi-continuous. Assume that the partitioning is exhaustive and that the deletion by infeasilibity rule is certain in the limit. Let $\{X_{k_q}\}$ be an infinitely decreasing sequence of successively refined nodes and suppose there exists $Q > 0$ so that for all nodes X_{k_q} with $q \geq Q$ all discontinuous factors appearing in the factorable representation of f and $\mathbf{g}$ have been branched on or are uniquely determined. Introduce ς_q and $Y_{\varsigma_q}(X_{k_q})$ as defined in Remark 7.2. Denote as f^{k_q} and $\mathbf{g}^{k_q}$ the functions obtained from f and $\mathbf{g}$ by branching according to ς_q. Let convex relaxations of f^{k_q} and $\mathbf{g}^{k_q}$ on X_{k_q} be calculated using standard McCormick relaxations. Furthermore, assume all remaining univariate functions appearing in the factorable representation*

of f and $\mathbf{g}$ are Lipschitz. Consider lower bounds obtained from

$$\beta(X_{k_q}) = \inf\{\check{f}^{k_q}(\mathbf{x}) : \mathbf{x} \in Y_{\varsigma_q}(X_{k_q}), \check{\mathbf{g}}^{k_q}(\mathbf{x}) \leq \mathbf{0}\}$$

Then, this bounding operation is strongly consistent.

Proof. Let $\varsigma = \varsigma_Q$ and note that $\varsigma_i = \{0, 1\}$, $\forall i \in I$. Exhaustive partitioning of the nodes implies that $[\mathbf{x}^*, \mathbf{x}^*] = \bigcap_{q=1}^{\infty} X_{k_q}$ and certainty in the limit of the deletion by infeasibility rule implies that $\mathbf{x}^* \in E$. Lemma 7.3 implies that the branch chosen by ς coincides with the evaluation of f and $\mathbf{g}$ at $\mathbf{x}^*$.

Thus, the assumptions of Lemmas 7.1 and 7.2 are met and it follows that

$$\lim_{q\to\infty} \beta(X_{k_q}) = \inf_{\mathbf{x}\in\bar{X}\cap E} f(\mathbf{x}). \qquad \square$$

Assumption 7.1. Assume that for every $\varepsilon > 0$, there exists a Q such that for all $q > Q$, $\alpha_{k_q} \leq f(\mathbf{x}^*) + \varepsilon$ where $[\mathbf{x}^*, \mathbf{x}^*] = \bigcap_{q=1}^{\infty} X_{k_q}$.

Theorem 7.3. *[cf. 88, Lemma IV.5] Suppose that f is lower semi-continuous, Assumption 7.1 holds and the partitioning is exhaustive. Furthermore, assume that for every infinitely decreasing sequence $\{X_{k_q}\}$ of successively refined nodes, there exists $\mathbf{x}_{k_q} \in S_{k_q}$ that satisfies $\mathbf{x}_{k_q} \in X_{k_q} \cap E$. Assume that for every $\varepsilon > 0$, there exists a Q such that for all $q > Q$, $\alpha_{k_q} \leq f(\mathbf{x}^*) + \varepsilon$ where $[x^*, \mathbf{x}^*] = \bigcap_{q=1}^{\infty} X_{k_q}$. Then, every strongly consistent lower bounding operation yields a consistent bounding operation.*

Proof. Since the partitioning is exhaustive and the bounding operation is strongly consistent, it follows that $\{X_{k_q}\} \to [\mathbf{x}^*, \mathbf{x}^*]$ where $\mathbf{x}^* \in E$ and $\lim_{q\to\infty} \beta(X_{k_q}) = \inf_{\mathbf{x}\in\bar{X}\cap E} f(\mathbf{x})$. Assumption 7.1 implies that $\lim_{q\to\infty} \alpha_{k_q} = f(\mathbf{x}^*)$ so that $\lim_{q\to\infty}(\alpha_{k_q} - \beta(X_{k_q}) = 0$. $\square$

Remark 7.4. Assumption 7.1, which is necessary to establish the assertion of Theorem 7.3, states that it is possible to identify feasible solutions with objective function value arbitrarily close to the optimal solution on the given node once the nodes become small enough. This is a much stronger assumption if f is not continuous.

7.1.3 Certainty in the limit of the deletion by infeasibility rule

Theorem 7.2 assumed that the deletion by infeasibility rule is certain in the limit. It is necessary to argue that branching on discontinuous factors will still allow fathoming infeasible nodes. Aside from infeasibility due to constraint violation, i.e., $E \cap \tilde{X} = \emptyset$, it is also necessary to remove nodes on which the discontinuous branch does not coincide with the function evaluation.

Lemma 7.4. *Assume that the partitioning is exhaustive. Let $\{X_{k_q}\}$ be an infinitely decreasing sequence of successively refined nodes and suppose there exists $Q > 0$ so that for all nodes X_{k_q} with $q \geq Q$ all discontinuous factors appearing in the factorable representation of f and $\mathbf{g}$ have*

been branched on or are uniquely determined. Introduce I, $\boldsymbol{\varsigma} = \boldsymbol{\varsigma}_Q$ *and* $Y_{\boldsymbol{\varsigma}}(X_{k_q})$ *as defined in Remark 7.2. Define*

$$\tilde{Y}_{\boldsymbol{\varsigma}}(\tilde{X}) = \left\{\mathbf{x} \in \tilde{X} : \exists i \in I : \overline{w}_i(\tilde{X}) \leq 0 \text{ if } \varsigma_i = 1 \text{ or } \underline{w}_i(\tilde{X}) > 0 \text{ if } \varsigma_i = 0\right\}. \tag{7.7}$$

It follows that $Y_{\boldsymbol{\varsigma}}(\bar{X}) \cup \tilde{Y}_{\boldsymbol{\varsigma}}(\bar{X}) = \bar{X}$ *and* $Y_{\boldsymbol{\varsigma}}(\bar{X}) \cap \tilde{Y}_{\boldsymbol{\varsigma}}(\bar{X}) = \emptyset$ *where* $\bar{X} = \bigcap_{q=1}^{\infty} X_{k_q}$.

Proof. Note that $\varsigma_i = \{0, 1\}$, $\forall i \in I$ when $q \geq Q$. Exhaustive partitioning implies that $\bar{X} = [\mathbf{x}^*, \mathbf{x}^*]$.

Suppose that $\mathbf{x}^* \notin Y_{\boldsymbol{\varsigma}}(\bar{X})$. Thus, there exists $i \in I$ so that $\underline{w}_i(\bar{X}) \leq 0$ if $\varsigma_i = 1$ or $\overline{w}_i(\bar{X}) > 0$ if $\varsigma_i = 0$. Since $\bar{X}$ is a degenerate interval, the interval extensions evaluated on $\bar{X}$ are exact so that $\underline{w}_i(\bar{X}) = \overline{w}_i(\bar{X}) \leq 0$ if $\varsigma_i = 1$ or $\overline{w}_i(\bar{X}) = \underline{w}_i(\bar{X}) > 0$ if $\varsigma_i = 0$. Hence, $\mathbf{x}^* \in \tilde{Y}_{\boldsymbol{\varsigma}}(\bar{X})$.

Suppose that $\mathbf{x}^* \in Y_{\boldsymbol{\varsigma}}(\bar{X})$. Then, $\underline{w}_i(\bar{X}) > 0$ if $\varsigma_i = 1$ or $\overline{w}_i(\bar{X}) \leq 0$ if $\varsigma_i = 0$ for all $i \in I$. It follows immediately that $\overline{w}_i(\bar{X}) > 0$ if $\varsigma_i = 1$ or $\underline{v}_i(\bar{X}) \leq 0$ if $\varsigma_i = 0$ for all $i \in I$. Hence, $\mathbf{x}^* \notin \tilde{Y}_{\boldsymbol{\varsigma}}(\bar{X})$.

Thus, the assertion follows. □

Corollary 7.2. Suppose that a node $\tilde{X}$ is fathomed when $\tilde{X} = \tilde{Y}_{\boldsymbol{\varsigma}}(\tilde{X})$ or when $D \cap \tilde{X} = \emptyset$. This is deletion by infeasility rule is certain in the limit.

7.2 Implementation details

Remark 7.5. For a practical implementation, it will be necessary to introduce a small nonnegative parameter ϵ and consider the set

$$Y'_{\boldsymbol{\varsigma}}(\tilde{X}) = \{\mathbf{x} \in \tilde{X} : \underline{w}_i(\tilde{X}) \geq \epsilon \text{ if } \varsigma_i = 1, \forall i \in I \text{ and } \overline{w}_i(\tilde{X}) \leq -\epsilon \text{ if } \varsigma_i = 0, \forall i \in I\}$$

instead of $Y_{\boldsymbol{\varsigma}}(\tilde{X})$ so that a practical deletion by infeasibility rule is certain in the limit. While this cuts off (small) parts of the feasible set, it ensures that $\tilde{X} \backslash Y'_{\boldsymbol{\varsigma}}(\tilde{X})$ is guaranteed to be feasible in the limit. Otherwise, it is not possible in a practicable implementation to delete infeasible nodes with certainty as each element of a convergent sequence may be feasible whereas the limit point is not.

Remark 7.6. Note that the branching heuristic is responsible for ensuring that eventually all discontinuous factors have been branched on (or are uniquely determined so that branching is not necessary, see Remark 7.1).

This functionality is added to MC++ [40], an open-source library that provides objects to construct McCormick relaxations of factorable functions through operator overloading. Three `static` private members are added to the class, a counter i and two binary vectors $\mathbf{p}$ and $\mathbf{q}$. i stores the number of calls to ψ, $\mathbf{p}$ stores if a discontinuous factor has been fixed ($p_i =$ `T`) and $\mathbf{q}$ stores the branch (i.e, $q_i =$ `T` when $\psi = 1$ and $q_i =$ `F` when $\psi = 0$); it is undefined if $p_i =$ `F`. Before a factorable function is evaluated, the counter must be reset and the integer vectors are passed to the class. Public member functions are provided for

these tasks. During the evaluation of the factorable function, the counter is incremented each time ψ is executed. When ψ is called and there are no constraints on its value, i.e., $p_i = \texttt{F}$, but the interval bounds indicate that only one branch is active, i.e., $\underline{v}_j > 0$ or $\overline{v}_j \leq 0$, p_i and q_i are updated accordingly. If $p_i = \texttt{T}$ then constraints on the assignment were set and q_i determines the returned value. Else $p_i = \texttt{F}$ and the relaxation are evaluated as detailed in Theorem 6.5. After the factorable function is evaluated, the value of the vectors can be obtained through additional public member functions, which is important when $\mathbf{p}$, $\mathbf{q}$ might have changed, i.e., when relaxations are evaluated on a host set for the first time and some discontinuous factors are uniquely determined on the host set.

Lastly, an exception is used to handle the case when $p_i = \texttt{T}$ and the value of q_i contradicts the branch choosen according to $\underline{v}_j$ or $\overline{v}_j$, e.g., $q_i = \texttt{F}$ and $\underline{v}_j > 0$. This exception has to be caught by the user. It indicates that the lower bounding problem on this node is infeasible and the node can hence be fathomed. Recall Remark 7.5 which said that a practical implementation needs to modify the feasibility test which checks for the correct branch of the discontinuous function. Here, the parameter is set to $\epsilon = 10^{-6}$.

7.3 Case studies

In this section, two case studies will be presented. First, Example 6.3, the motivating example, will be revisited. Then, a more complicated case study will be considered.

Before the case studies are discussed, a few remarks regarding methods to solve the convex minimization problem to find $\beta(\cdot)$, heuristics to determine feasible points, and hence α_k, branching and node selection heuristics, and the utilized tolerances are necessary.

Several methods are compared for solving the convex problem at iteration k on node X_k and finding a feasible point.

Method 1 The bound from interval arithmetic, $\underline{f}$, is used as $\beta(X_k)$. The midpoint of the interval X_k is added to S_k. This procedure yields very efficient lower bounds at the expense of tightness.

Method 2 An affine approximation of the convex relaxation of the objective function is constructed sequentially. First, a subgradient of $\check{f}$ is evaluated at the midpoint of X^l and an affine relaxation of f is thus constructed. Combined with the interval bound, $\underline{f}$, CPLEX is used to find a minimum of the affine relaxations. A subgradient of $\check{f}$ is evaluated at this solution, another affine relaxation is added and CPLEX is used to solve this problem. To balance efficiency and accuracy, a total of five minimization problems are solved with CPLEX. The last solution found is reported as $\beta(X_k)$. The last point found by CPLEX is added to S_k.

Method 3 A bundle solver [113] with bundle size 15 is used to find the minimum of the convex relaxation of the objective function. The point returned by the bundle solver is added to S_k.

k	$\underline{x}_k$	$\overline{x}_k$	$\mathbf{q}^k$	$\beta(X_k)$	N_k
1	-1.0	1.0	$(-,-)$	-1	2
2	-1.0	0.0	$(-,-)$	0	1
3	0.0	1.0	$(-,-)$	-1	2
4	0.0	0.5	$(-,-)$	-1	3
5	0.5	1.0	$(-,-)$	0	2
6	0.0	0.25	$(-,-)$	-1	3
7	0.25	0.5	$(-,-)$	0	2
8	0.0	0.125	$(-,-)$	-1	3
9	0.125	0.25	$(-,-)$	0	2
10	0.0	0.125	$(\mathtt{F},-)$	$+\infty$	1
11	0.0	0.125	$(\mathtt{T},-)$	0	0

Table 7.1: Nodes visited and bounds calculated for the motivating example

The best bound heuristic is used to determine the next node and the absolute diameter heuristic is used to select on which variable to branch. Furthermore, another heuristic is necessary to decide when to branch on discontinuous factors. Currently, the algorithm branches on first discontinuous factors that have not been branched on if $w(X_k) < \epsilon_d w(X)$ where $\epsilon_d = 0.1$. Lastly, a node $\tilde{X}$ is deleted at iteration k if either $\alpha_k - \beta(\tilde{X}) < 10^{-5}$ or $\alpha_k - \beta(\tilde{X}) < 10^{-1}\beta(\tilde{X})$. At most 200,000 iterations are undertaken.

7.3.1 Motivating example revisited

It is instructive to study the motivating example, i.e., $f(x) = \psi(x) - \psi(x)$, more closely. Consider the factorable representation of f as $v_1 = x$, $v_2 = \psi(v_1)$, $v_3 = \psi(v_1)$, and $f = v_4 = v_2 - v_3$. In this formulation there are two discontinuous factors to branch on; p_1, q_1 correspond to v_2 while p_2, q_2 correspond to v_3.

Recall that $f(x) = 0$ for any x. Hence, any feasible point yields an optimal solution and $\alpha_k = 0$ for any k. In Table 7.1 the current node X_k is characterized by $\underline{x}_k$ and $\overline{x}_k$ for each iteration and the state of $\mathbf{q}^k$ is given where $q_i^k = -$ implies that $p_i^k = \mathtt{F}$. Also, the lower bound on the node and the number of nodes remaining in the stack N^k are stated. The following observations can be made. First, any node with $\underline{x} > 0$ or $\overline{x} \leq 0$ can be deleted as the lower bound on such nodes is at least as great as the optimal solution. Nodes of the type $[0, \varsigma]$ remain undeleted since $\beta([0, \delta]) = -1$. In particular, these are refined to obtain nodes $[0, \frac{\delta}{2}]$ and $[\frac{\delta}{2}, \delta]$, the latter of which can be deleted again.

Also, it is important to point out that the heuristic that determines when to branch on discontinuous variables is important here. As soon as the first discontinuous variable is branched on, the bounds can be brought up and the algorithm converges. This example has the advantage that every other node can be deleted immediately thus keeping the number of nodes in the stack from growing. In general, it may be possible that a large number of undeleted nodes has been generated on which branching on discontinuous

factors occurs.

Lastly, it should be noted that each of the methods listed at the beginning of this section visit the same sequence of nodes.

7.3.2 Parameter estimation with embedded dynamic model

The second example considers a parameter estimation problem with embedded dynamics described by a discrete-time hybrid system. Consider a continuous stirred-tank reactor (CSTR) with residence time $\tau = 5$ in which a reaction that is characterized by Michaelis–Menten kinetics converts substrate (S) into product (P) catalyzed by an enzyme (E). The dynamics are discretized using $N = 50$ equally spaced time points with $t_{i+1} = t_i + \Delta t$, $i \in I = \{1, \ldots, N\}$ where $\Delta t = 0.02$. The feed stream to the reactor contains substrate with concentration $c_{S_0} = 2$. The enzyme is washed out of the reactor and it also deactivates with rate $a(T - T_0)^2$ where $a = 0.001$ and $T_0 = 310$ are empirical parameters. The outlet concentration of P and E, $c_{E,i}$ and $c_{P,i}$, respectively, are measured when $i \in I_s = \{10, 20, 30, 40, 50\}$. When the concentration drops below a threshold, i.e., if $c_{E,i+1} < c_E^0$ for $i \in I_{\text{sample}}$, additional enzyme is added to the reactor boosting enzyme concentration by Δc_E where $c_E^0 = 0.05$ and $\Delta c_E = 0.15$. The measured concentration of P is used to determine two unknown parameters, k_∞ and T. The scaled activation energy $\frac{E}{k} = 300$ and reaction rate parameter $K = 1$ are known. Experimentally determined effluent concentrations $c_{P,i}^{\text{exp}}$, $i \in I_{\text{sample}}$ are available to estimate the parameters, see Table 7.2. Initial values for the concentrations are $c_{S,1} = c_{S_0}$, $c_{P,1} = 0$, $c_{E,1} = 0.1$ and bounds on the variables are $k_\infty \in [70, 140]$, $T \in [290, 320]$. The optimization problem can be written as follows:

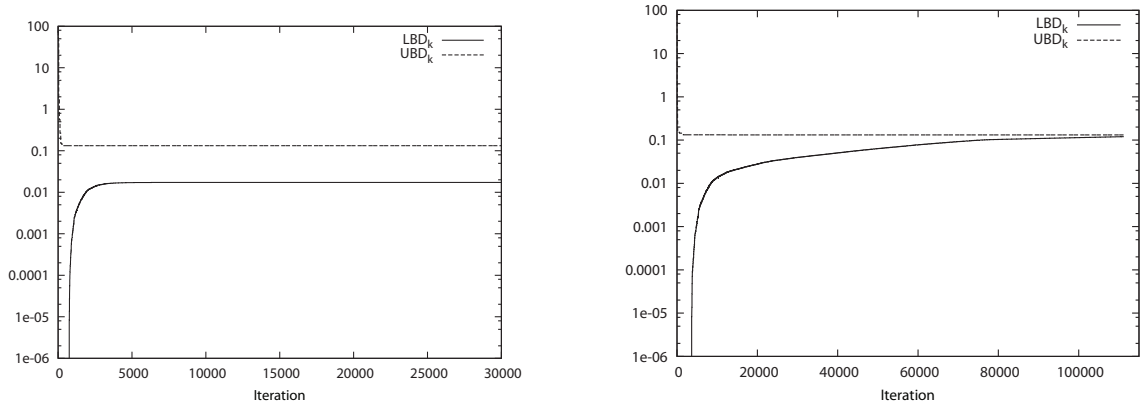
$$
\begin{aligned}
\min_{k_\infty, T} \quad & 1000 \sum_{i \in I_{\text{sample}}} \left(c_{P,i} - c_{P,i}^{\text{exp}} \right)^2 \\
\text{s.t.} \quad & c_{S,i+1} = c_{S,i} + \Delta t g_S(c_{S,i}, c_{S,i}, k_0), \\
& c_{P,i+1} = c_{P,i} + \Delta t g_P(c_{P,i}, c_{S,i}, c_{S,i}, k_0), \\
& \tilde{c}_{E,i} = c_{E,i} + \Delta t g_E(c_{E,i}, T), \\
& c_{E,i+1} = \tilde{c}_{E,i} + \begin{cases} \Delta c_E & \text{if } i \in I_s, \tilde{c}_{E,i} < c_E^0, \\ 0 & \text{otherwise,} \end{cases} \\
& k_0 = k_\infty \exp\left(-\frac{E}{kT} \right).
\end{aligned}
$$

where

$$
\begin{aligned}
g_S(c_{S,i}, c_{E,i}, k_0) &= (c_{S_0} - c_{S,i})\tau - \frac{k_0 c_{E,i} c_{S,i}}{K + c_{S,i}}, \\
g_P(c_{P,i}, c_{S,i}, c_{E,i}, k_0) &= -c_{P,i}\tau + \frac{k_0 c_{E,i} c_{S,i}}{K + c_{S,i}}, \\
g_E(c_{E,i}, T) &= -c_{E,i}\tau - a(T - T_0)^2.
\end{aligned}
$$

i	$c_{P,i}^{\exp}(t_i)$
10	0.16262
20	0.33426
30	0.43722
40	0.22993
50	0.34085

Table 7.2: Experimentally measured concentration of product in the reactor effluent



(a) without branching

(b) with branching

Figure 7.1: Comparing the convergence behaviour of the upper and lower bounds without and with branching on discontinuous factors for the parameter estimation problem

The discontinuity is introduced in this problem by the decision if enzyme needs to be added to the reactor which depends on the current enzyme concentration. If this addition is triggered, the concentration c_E jumps and consequently impacts subsequent substrate and product concentrations. The objective function is shown in Figure 7.2 and clearly exhibits multiple discontinuities and local minima that are not global minima.

The problem can be cast as an algorithm that calculates the objective given $\mathbf{x} = (k_\infty, T)$. Events are represented as discontinuities using ψ. Bounds on the concentrations, which are known from physical considerations, are utilized to strengthen the relaxations, i.e., $c_{P,i} \geq 0$, $c_{S,i} \in [0, C_{S_0}]$, $c_{E,i} \in [0, c_E^0 + \Delta c_E]$.

Consider Figure 7.1(a) which demonstrates that this problem also exhibits a finite convergence gap. However, the identification of its source is not as simple as in the previously considered example. In order to converge the bounds, branching on discontinuous factors is used in this problem, too. Note that there are 5 discontinuous factors on which branching can occur. Figure 7.1(b) indicates that convergence can be achieved after allowing branching on discontinuous factors. Additionally, the method to tighten bounds discussed

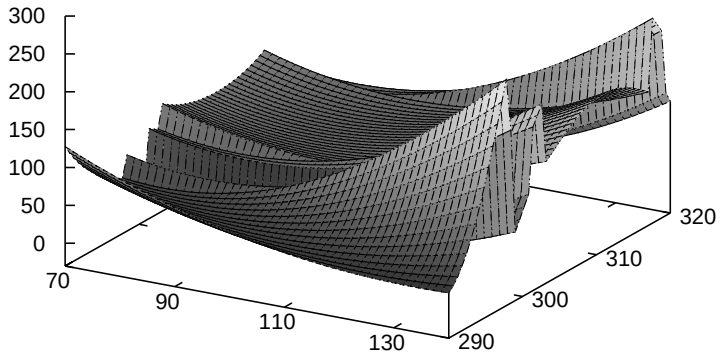


Figure 7.2: Objective function of parameter estimation problem with embedded discrete-time hybrid system

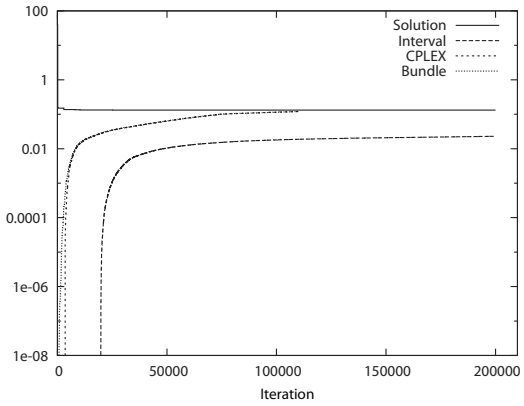


Figure 7.3: Convergence behavior of lower bounds where interval and relaxations are used to compute bounds

in Appendix A is used.

Detailed results comparing the different methods are reported in Table 7.3. As also shown in Figure 7.3, solving the convex relaxations with Method 2 and 3 gives very similar results though Method 2 is nearly twice expensive, see Table 7.3. Method 1 was not able to converge the lower bound in the iterations alloted though an optimal solution has been identified.

7.4 Discussion

In this section, special focus will be put on the impact of allowing branching on discontinuous factors, deciding when a node can be fathomed and how to branch and a remark regarding different methods to find the lower bounding problem.

Method	# LBPs	# UBPs	Runtime [s]	f^*	$\mathbf{x}_{\min}$
1	$> 179,602$	$> 172,537$	> 294.3	0.1321	(97.86, 299.6)
2	96,838	54,960	619.9	0.1321	(97.83, 299.6)
3	97,248	55,179	337.5	0.1321	(97.89, 299.6)

Table 7.3: Comparison of different methods for the parameter estimation problem with embedded discrete-time hybrid system.

Importance of branching on discontinuous factors Branching on discontinuous variables as introduced in Section 7.1 is a method to reduce the convergence gap. It replaces a node on which f and $\mathbf{g}$ cannot be guaranteed (using interval arithmetic) to be continuous with two nodes by fixing a discontinuous factor to either branch of the discontinuous function. Initially, this may loosen the relaxation as argued in Corollary 7.1. However, once all discontinuous factors have been branched on or can be guaranteed to be continuous, the resulting factorable function is continuous. Now, standard results from interval analysis guarantee convergence of relaxation and interval bounds to the constructed continuous function.

In contrast to a formulation with binary variables [e.g., 19, 168], using a discontinuous representation has several advantages. First, calculating an upper bound, and hence finding a feasible solution, does not require fixing all binary variables to integer values. For a box constrained problem, evaluating the objective at any point in the box yields a valid, though not necessarily tight, upper bound. Second, to converge the bounds one may not have to branch on all discontinuous factors in a practical problem (though this cannot be ruled out in principle) as shown in the first case study in Section 7.3.1. In particular, this formulation can take advantage of interval information to decide if discontinuous factors are already uniquely determined on the current host set and eliminate the possibility of branching on it subsequently.

Infeasibility test As already insinuated in Remark 7.5, the test for infeasibility needs to be modified. This modification has analogues in nonlinear programming. Consider a general constraint $g(\mathbf{x}) \leq 0$. Often, practical implementations that consider such constraints will also introduce a small parameter which specifies the tolerance to which the constraint need to be satisfied. For example, an algorithm based on active-set strategy needs to determine which constraints are active and then will also need to solve a nonlinear equation. Both steps can only be carried out numerically to within some small tolerance.

The necessary property of the deletion by infeasibility rule is given in Definition 6.9. However, the statement of the test for infeasibility as given in Remark 7.2 does not have the required property as first pointed out in Remark 7.5. It shall be reiterated here that this modification is typically for actual implementations.

A clear consequence of this modification is the following. Consider $h : X \to \mathbb{R}$ as a token of f or g_j. Suppose h is discontinuous at $\mathbf{x}^\ddagger \in X$ so that $\exists \varepsilon' > 0$ and for any $\varepsilon' > \varepsilon > 0$ there does not exist a $\delta > 0$ so that $f(\mathbf{x}) - f(\mathbf{x}^\ddagger) < \varepsilon$ for all $\mathbf{x} \in X$ with $\|\mathbf{x} - \mathbf{x}^\ddagger\|_2 < \delta$.

ϵ_d^{-1}	# LBPs	# UBPs
4	99,003	56,321
8	97,274	55,196
10	97,248	55,179
16	95,378	53,882
32	92,789	51,949
64	89,168	48,931
128	87,826	48,077
256	89,607	48,192
512	95,804	48,556
1024	118,667	59,366

Table 7.4: Influence of heuristic for discontinuous branching on convergence of second case study using Method 3

Thus, h at $(\mathbf{x}^\ddagger)$ is lower than h at any point in its vicinity. In particular, it is lower by a non-infinitesimal amount. While it may be difficult to identify such a (local) minimum with numerical methods in general, the modification of the deletion by infeasibility rule can remove $\mathbf{x}^\ddagger$ from the feasible region. Thus, the algorithm would converge to a suboptimal point but report that it found an optimal solution. However, for practical problems, one would not be interested in optimal solutions that are not valid in a neighborhood of the optimum as these are typically impossible to realize in practice anyway.

Branching heuristic As demonstrated with the first case study, the branching heuristic that decides when to branch on discontinuous variables can have a large impact on the convergence behaviour.

On the one hand, branching on discontinuous variables early, i.e., at a high level in the branch and bound tree, will spawn only a small number of new nodes and the functions that are relaxed are continuous early on. On the other hand, branching early will reduce the possibility that interval bounds can rule out one branch of the discontinuous factor and hence eliminate the need to branch on this factor. However, branching at deeper levels of the branch and bound tree will require that the same discontinuous factor will be branched on multiple times and thus enlarging the number of nodes to visit considerably. This is especially important as the relaxations right after branching on a discontinuous factor will result in looser bounds on at least one node than on the parent node. In Table 7.4, it is shown how the number of considered lower and upper bounding problems change when ϵ_d is modified. Note that multiplying ϵ_d by a factor of two means that branching on discontinuous factors occurs when the width of the host set is twice as large. The sensitivity study shows the trade-off discussed before. However, it is too early to generalize these results. In general, the discontinuity branching heuristic will need to balance this trade-off carefully.

Methods for lower bounding problem A few remarks regarding Figure 7.3 are in order. First, it demonstrates the value derived from convex relaxations over bounds from interval arithmetic. Secondly, it shows that the bundle solver is able to provide tighter lower bounds initially although slightly more nodes are visited overall. Also, as reported in Table 7.3, the calls to CPLEX introduce significant overhead as evident by the increase in runtime by 84% when compared to the bundle solver.

7.5 Conclusion

In Section 7.1, a method was introduced to overcome this convergence problem by allowing branching on discontinuous factors. Theoretical results are established to guarantee that branching on discontinuous factors continues to provide valid bounds. Furthermore, it was established that the bounds convergence assuming the the deletion by infeasibility rule is certain in the limit. Additionally, a result was given that allows to tighten bounds without using dual information. Lastly, two case studies demonstrate that convergence can be achieved.

Chapter 8

Conclusion

This thesis contains original contributions to the field of global optimization as well as process design and heat integration.[1] Optimization is a key activity in any engineering discipline. In chemical engineering, in particular, process models are often nonlinear and nonconvex, e.g., due to the presence of multiple components, nonlinear thermodynamic models and kinetic mechanisms. Deterministic global optimization algorithms are capable of solving such process models even in the presence of non-optimal local solutions. Unfortunately, the worst-case runtime of all known algorithms scales exponentially with the problem dimension. To circumvent this behavior, it is suggested to reduce the number of model variables visible to the optimizer, e.g., by using equality constraints to solve for some of the variables. The reduced-space formulation leads to several complications that were addressed here: the optimum of the resulting problem formulation is more likely to be unconstrained potentially worsening the cluster problem; the information contained in the constraints needs to be better exploited in the construction of bounds and relaxations; standard regularity assumptions (differentiability and continuity) of the resulting formulation cannot always be guaranteed.

The resulting reduced-space problem formulation is more prone to the cluster problem [54], the phenomenon that global optimization algorithms visit a large number of boxes in the immediate vicinity of unconstrained global (and near global) solutions. In this thesis, the previous analysis in [129] was revisited and improved. The importance of second-order convergent bounding methods was confirmed and it was also shown that the tightness of the bounding method significantly impacts the incidence of this phenomenon.

Based on the representation of the constraints as a directed acyclic graph, a method analogue to constraint propagation for convex and concave relaxations was proposed that extends a method for interval bounds [174]. The variables are partitioned into independent and dependent variables. First, relaxations of the constraints are computed using generalized McCormick relaxations. Next, these relaxations are intersected with the constraint values. Then the graph is traversed in reverse order and the computations are inverted in some sense. This technique provides relaxation of the set-valued mapping from independent to dependent variables implied by the constraints. These relaxations were shown to improve the standard McCormick relaxations of the feasible set. Compared to

[1] A formulation for the combined process optimization and heat integration for processes at subambient conditions was proposed and a pinch operator for streams with non-constant heat capacity was introduced. These are documented in Appendices B and C.

prior methods [e.g., 164], existence and uniqueness of an implicit function implied by the equality constraints is not presumed. Additionally, the information contained in inequality constraints can also be incorporated in this procedure and repeated application of the method can further improve the obtained relaxations.

Second-order convergent bounding methods are essential in mitigating the cluster problem. It has been shown that standard McCormick relaxations are second-order convergent [34]; this is true for generalized McCormick relaxations only if all relaxations appearing in the arguments are also second-order convergent. When constructing relaxations of implicit functions implied by a system of nonlinear equations, interval bounds obtained from parametric interval-Newton methods are used to initialize generalized McCormick relaxations [e.g., 164]. It was argued in this thesis that these bounds are first-order convergent only and a second-order convergent bounding procedure based on the sensitivities of the system was discussed.

While the existing theory for McCormick relaxations required continuous functions, an extension to a class of discontinuous functions was proposed. Previously, binary variables [e.g., 168] or equilibrium constraints [e.g., 20] were employed to model discontinuous behavior in systems, which leads to a, sometimes significant, increase in the problem dimension. The properties of the obtained relaxations were analyzed and, under certain assumptions, convergence was established. Furthermore, branch-and-bound theory was revisited and extended to this case. A further extension was presented to guarantee convergence of the relaxation in a more general setting by allowing branching on the discontinuous factors.

8.1 Future work

It is clear that bounding methods from the class of centered forms possess the same convergence order as McCormick or αBB relaxations. Additionally, the former do not require the solution of an optimization problem to bound the range. On the other hand, bounding methods suffer from the wrapping effect when it comes to overestimating the feasible region; see Section 3.5. Though the wrapping effect supplies a plausible explanation why relaxations have been more successful in global optimization, extensive numerical studies could confirm this or, otherwise, provide new insights for future research. It is also a plausible conjecture that relaxations provide an easier avenue to perform domain reduction, which is essential in any practical global optimization implementation.

Child nodes in a branch-and-bound tree are closely related to the parent node and much information constructed for the parent node may be useful when solving the lower and upper bounding problem on the child node. At the same time, passing on too much of this information will significantly increase storage requirements of each node. It may be interesting to explore possibilities to enable warm-starting the optimization procedures in the bounding problems of the child node using information from the parent node. For example, when a bundle method is used to minimize the convex relaxation of the objective function, the bundle of the parent node should also be applicable to the child node.

The bundle solver [113] used to minimize nonsmooth convex relaxations in this thesis does not provide duality information, which can be a quite effective means in optimality-based domain reduction methods [e.g., 149]. If the current solver can be replaced with a different implementation that does provide duality information for the solution, more effective domain reduction methods than those discussed in Appendix A are possible.

The auxiliary variable method as implemented in BARON relies on the linearization of the convex and concave relaxations of the constraints since it uses LP algorithms in the lower bounding problem. While LPs can be solved more reliably and, often, also more efficiently than the original convex program, the resulting bounds are weaker. It would be interesting to study this trade-off for McCormick relaxations numerically as it is difficult to obtain theoretical results for the linearization.

The bounding method presented in Chapter 5 could be combined with either the reverse McCormick propagation described in Chapter 4 or the method introduced in [164] to construct relaxations of implicit functions.

While the bounding method presented in Chapter 5 is second-order convergent, first case studies exhibit a clear dependence of the convergence order pre-factor on the number of dependent variables. It is possible that this is due to the particular nature of the case studies as systems of equations obtained by discretizing ODEs. On-going work in the group focusing on the convergence order of McCormick relaxations of ODEs indicate that the pre-factor grows rapidly as the ODE is integrated to longer times. Therefore, the behavior of the bounding method is not unexpected. However, more studies may be warranted. In general, it is expected, but not yet studied in detail, that the convergence order pre-factors of the interval bounds and relaxations of factorable function grow with the length of its computational sequence.

In similar spirit, when sparsity and block decomposition can be exploited as in some of the case studies in Chapter 5, it is conceivable that interval bounding and McCormick relaxations benefit differently. The interval methods presented there have a clear dependence on the size of the Jacobian matrix, which in turn depends on the dependent variables $\mathbf{x}$. In other words, the block structure allows to solve many smaller systems instead of one large one leading to better bounds. On the other hand, it is anticipated that the tightness of McCormick relaxations depends on the length of the computational sequence. In this case, using reverse McCormick propagation on the complete, but very sparse system may be more beneficial. This warrants more investigation in the future.

In order to develop the relaxation methods for discontinuous functions further, the rate of convergence of the relaxations must be increased. While the extension presented in Chapter 7 can guarantee convergence even when Assumption 6.2 does not hold, the rate of convergence can be low and, consequently, full space problem formulations are currently more efficient. Also, the constraint propagation technique developed in Chapter 4 could be used in combination with branching on the discontinuous factors.

Appendix A

Domain reduction using subgradients

An important feature of efficient branch-and-bound procedures for global optimization are methods to reduce the search space by cutting regions that are known not to contain an optimal solution, also known as range reduction [91, 148, 149, 166].

In the absence of dual information that is used in optimality-based range reduction techniques [148, 149], one can utilize subgradient information.

Theorem A.1. *Suppose UBD is a valid upper bound of* (1.1) *on C,* $X = [\underline{\mathbf{x}}, \overline{\mathbf{x}}] \in \mathbb{IC}$, $\tilde{\mathbf{x}} \in X$, $\check{f}$ *is a convex relaxation of* f *on* X *and* $\boldsymbol{\sigma} \in \partial \check{f}(\tilde{\mathbf{x}})$. *Let* $i \in \{1, \ldots, n\}$ *so that* $\sigma_i \neq 0$. *Then, the bounds can be updated as follows:*

$$\overline{x}_{i,new} = \max\{\underline{x}_i, \tilde{x}_i + \frac{\omega_i}{\sigma_i}\} \quad \text{if } \sigma_i > 0,$$
$$\underline{x}_{i,new} = \min\{\overline{x}_i, \tilde{x}_i + \frac{\omega_i}{\sigma_i}\} \quad \text{if } \sigma_i < 0,$$

where

$$\omega_i = UBD - \check{f}(\tilde{\mathbf{x}}) - \sum_{j \neq i} \min\{\sigma_j(\underline{x}_j - \tilde{x}_j), \sigma_j(\overline{x}_j - \tilde{x}_j)\}.$$

Proof. Since $\check{f}$ is a convex relaxation of f on X, it holds for any $\mathbf{x} \in X$ that

$$f(\mathbf{x}) \geq \check{f}(\mathbf{x}) \geq \check{f}(\tilde{\mathbf{x}}) + \boldsymbol{\sigma}^{\mathrm{T}}(\mathbf{x} - \tilde{\mathbf{x}}).$$

Thus, adding the constraint

$$UBD \geq \check{f}(\tilde{\mathbf{x}}) + \boldsymbol{\sigma}^{\mathrm{T}}(\mathbf{x} - \tilde{\mathbf{x}})$$

to (1.1) will not change the solution of the optimization problem. In particular,

$$UBD - \check{f}(\tilde{\mathbf{x}}) - \sum_{j \neq i} \sigma_j(x_j - \tilde{x}_j) \geq \sigma_i(x_i - \tilde{x}_i).$$

It holds for all $j \in \{1, \ldots, n\}$ and $x_j \in [\underline{x}_j, \overline{x}_j]$ that

$$\sigma_j(x_j - \tilde{x}_j) \geq \min\{\sigma_j(\underline{x}_j - \tilde{x}_j), \sigma_j(\overline{x}_j - \tilde{x}_j)\}.$$

Hence, $\sigma_i(x_i - \tilde{x}_i) \leq \omega_i$ for all $x_i \in [\underline{x}_i, \overline{x}_i]$. If $\sigma_i > 0$, then $x_i \leq \tilde{x}_j + \frac{\omega_i}{\sigma_i}$. If $\sigma_i < 0$, then

$x_i \geq \tilde{x}_j + \frac{\omega_i}{\sigma_i}$. Lastly, the updated bounds need to satisfy $\underline{x}_{i,\text{new}} \leq \overline{x}_i$ and $\overline{x}_{i,\text{new}} \geq \underline{x}_i$ so that the assertion follows. □

Remark A.1. Theorem A.1 does not specify where to evaluate f and ∂f. As a first heuristic, one can use the solution returned by the lower bounding problem. However, if $\tilde{\mathbf{x}}$ is a optimal solution of the lower bounding problem on X, then σ may not contain a non-zero element, in which case the procedure will not yield any new information.

Remark A.2. To reduce computational effort at each iteration, one can choose to update only the bound that correspond to the dimension with the largest absolute subgradient, i.e., let $i \in \arg\max_i\{|\sigma_i|\}$. Note that this is a heuristic only.

Appendix B

Synthesis of heat exchanger networks at subambient conditions with compression and expansion of process streams[1]

B.1 Introduction

The design of energy efficient processes received a lot of attention during the energy crises of the 1970s, and has recently attracted new attention due to the current high cost of energy, as well as the new goal of reduced CO_2 emissions to mitigate global warming. Consequently, researchers have studied methodologies for the optimal design of heat exchanger networks [65, 72, 92, 93]. One of the most successful tools to optimize energy integration during process design is pinch analysis. By providing a rigorous lower bound for the utilities needed by a given process design, it serves as a guideline for achievable process integration during flowsheet synthesis [33, 111, 157, 158, 160].

The decomposition of the design process for chemical plants has been previously illustrated by the Onion Diagram, which indicates the levels of process design as well as the natural sequence of decisions. Commonly, at the core of the Onion Diagram is the Reactor System (R), followed by the Separation System (S), the Heat Recovery System (H) and the Utility System (U). In the first version of the Onion Diagram, however, Linnhoff et al. [111] did not include the utility level, but more important, they included Compression and Expansion (C&E) inside the Heat Recovery System (see Figure 1). This is interesting and highly relevant to the present appendix, since expanders and compressors play a significant role in the proposed methodology. Important "feedbacks" from outer to inner layers exist, however, which complicate the simplified flow of information in one direction indicated in Figure 1. One of these feedbacks is related to the interaction between setting the pressure of separation equipment, such as distillation columns and evaporators, and the design of the heat recovery system. By changing the pressure levels of such separators, the corresponding temperature levels of important (large duties) heat sources and sinks will change. This may have a significant impact on the scope for direct heat integration or heat pumping. Figure B.1 shows an extended version of the traditional Onion Diagram where compression and expansion of the process streams are taken to be separate operations

[1] This chapter, which has been published in *AIChE Journal*, 57(8):2090–2108, 2011, is joint-work with Audun Aspelund and Truls Gundersen.

isolated from the utility system. Use of the concepts underlying this extended Onion Diagram is a central part of the ExPAnD methodology presented by Aspelund et al. [9].

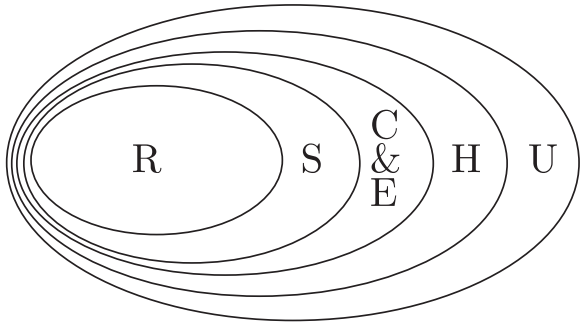


Figure B.1: Illustration of the natural sequence in process design (**R**eactor system, **S**eparation system, **C**ompression and **E**xpansion, **H**eat exchanger system, **U**tilities)

Pinch analysis (PA) covers the interface between the basic process (R, S) and the utility system (U) with special focus on the heat recovery system (H). This methodology has reached a mature level of industrial application over the years and has been successfully applied to improve heat recovery, to design better heat and power systems and utility systems, as well as in many other aspects of process design. An introduction to and overview with references to original research are given in the textbook of Smith [160]. The major limitation with this methodology is that only temperature is used as a quality parameter, thus neglecting pressure and composition. Over the last decades, PA has been a source of physical insights, which have led to advances in the synthesis of heat exchanger networks. Two thorough reviews of heat exchanger network synthesis (HENS) were published by Gundersen and Naess [72] and by Jezowski [92, 93]. More recently in [2002], Furman and Sahinidis [65] contributed a critical review and annotated bibliography of 461 papers on the design of HENs.

Exergy analysis (EA) [101] can be used in all stages of process synthesis (PS), and the advantage is the inherent capability of including all stream properties (temperature, pressure and composition); however, a limitation is its focus on the equipment units, rather than the flowsheet level. In addition, there is no obvious conversion from exergy to cost; in fact, there is often a conflict between reducing exergy losses and reducing cost. Nevertheless, Anantharaman et al. [4] tried to combine PA and EA in drawing so-called Energy Level Composite Curves. The new Energy Level parameter was proposed by Feng and Zhu [60]. Hom**v**sak and Glavi**v**c [85] suggested power availability curves to visualize the effect of pressure changes, which are not taken into account in the traditional composite curves of PA, in order to guide the appropriate placement of compressors and expanders.

Optimization techniques, usually referred to as mathematical programming (MP), are widely used in PS. One of the main challenges in their application is the fact that most problems in process design feature discrete decisions between process alternatives in addition to nonlinear process models as well as economic models with continuous variables.

Thus, these problems need to be cast as Mixed Integer Nonlinear Programs (MINLP). An inherent property of such problems, even for problems of relatively small size, is the difficulty to guarantee that the global optimal solution has been found. Simultaneous optimization and heat integration of chemical processes can be formulated using MP, and several authors have contributed ideas [10, 12, 55, 71, 77, 131–134, 179–181]. The reader is referred to the review by Furman and Sahinidis for additional references on the sequential and simultaneous synthesis of HEN [65].

Although there have been extensive efforts to optimize heat exchanger networks, very few papers have been published describing how the pressure of process streams can be manipulated in order to achieve more energy- and cost-effective processes. This is especially important in energy-intensive cryogenic processes, such as the liquefaction of natural gas or hydrogen, where the temperature of process streams is very sensitive to changes in pressure as the boiling or condensation temperature is a function of pressure. Furthermore, expansion or compression of process streams changes both temperature and pressure, and converts stream enthalpy into work or vice versa. For example, a pressurized stream can be expanded to produce both thermal (cold) exergy and work from pressure exergy. Earlier, efforts have been made to develop an Extended Pinch Analysis and Design procedure (ExPAnD) and study the Attainable Regions (AR) for expansion of process streams at subambient temperatures [6, 9]. However, these procedures rely on heuristics and a graphical interpretation of pressure exergy. It would be beneficial to formulate the problem using MP. Holiastos and Manousiouthakis [84] have shown the theoretical potential for moving the composite curves closer together by using ideal heat and power processes based on the second law of thermodynamics. This may provide interesting theoretical insights; however, such processes cannot be implemented as real engineered systems.

This appendix presents a process design tool that combines PA, EA and MP to find heat exchanger networks with minimal irreversibility by varying pressure levels of process streams. It is structured as follows. First the problem statement is given. Then the state space approach to modelling the problem is described providing a detailed formulation of the pressure operator, the pinch operator and the exergy operator. Two examples highlight the application of the formulation: First, a simple example with one hot stream and two cold streams is considered, where the pressure of one of the cold streams can be manipulated. Then the methodology is applied to achieve a better design in a novel process for liquefaction of natural gas using liquid CO_2 (LCO_2) and liquid inert nitrogen (LIN) as cold carriers [8].

B.2 Problem statement

The problem statement can be formulated as follows:

> *Given a set of process streams with a supply state (temperature, pressure and the resulting phase) and a target state, as well as utilities for power, heating and cooling; design a system of heat exchangers, expanders, pumps and compressors in such a way*

that an objective is minimized.

It should be emphasized that this problem definition is significantly more complex than the standard heat recovery problem in PA. First, the issue of soft target temperatures is now expanded to include also soft target pressures. Second, the thermodynamic process from the initial to the final state is not specified, and the change in temperature and pressure as well as phase may follow a large set of different routes. Third, the distinction between process streams and utilities, as well as between hot and cold streams, is no longer obvious. In fact, streams may change identity; for example, a cold process stream may temporarily change to a hot stream, and vice versa. Some process streams act like utilities by providing energy sources or sinks at temperatures outside the range spanned by the available utilities. Additionally, stream properties such as phase can be changed by manipulating the pressure. Finally, note that the actual problem considered will suggest the objective. Typically, it will correspond to some representation of operating cost, but an example will show that minimization of utility cost may be meaningless in some cases. Another possible choice for the objective is the exergy efficiency of the process.

In this appendix, the typical assumptions that are made in PA are used. Process streams are considered to have constant heat capacity, and pressure drops across heat exchangers are neglected. Since the heat capacity varies with temperature, the assumption of constant heat capacity may lead to significant error. This can be mitigated by splitting a stream into several piecewise segments with different heat capacities depending on the temperature interval. Recently, a formulation that allows for nonconstant heat capacities was presented in [77] where an empirical cubic correlation is used. It should be pointed out that this formulation is significantly more complex as it requires more variables and constraints. When combined with varying process conditions (flowrates, temperatures and pressures), the problem cannot be solved in reasonable time. Expansion and compression of streams are modelled as isentropic processes, while an isentropic efficiency factor is introduced to adjust for unavoidable losses in real processes. To model the thermodynamic behavior of the fluid as the pressure changes, any equation of state can be used in principle [135]; here, for simplicity, the ideal gas model is used.

B.3 Description of the process model

B.3.1 A state space approach for design of heat exchanger networks including compressors and expanders

The state space approach to mass and heat transfer network design was presented by Bagajewicz et al. [10]. The paper describes a way to divide the operations into mass and heat transfer. On a similar basis, the state space realization of a HEN and compressor/expander network is shown in Figure B.2.

The pinch operator locates the pinch point and thus infers the minimum utility requirements for the process streams. These are divided into two categories, fixed and variable. The pressure of fixed streams is constant, while it can be changed through expansion

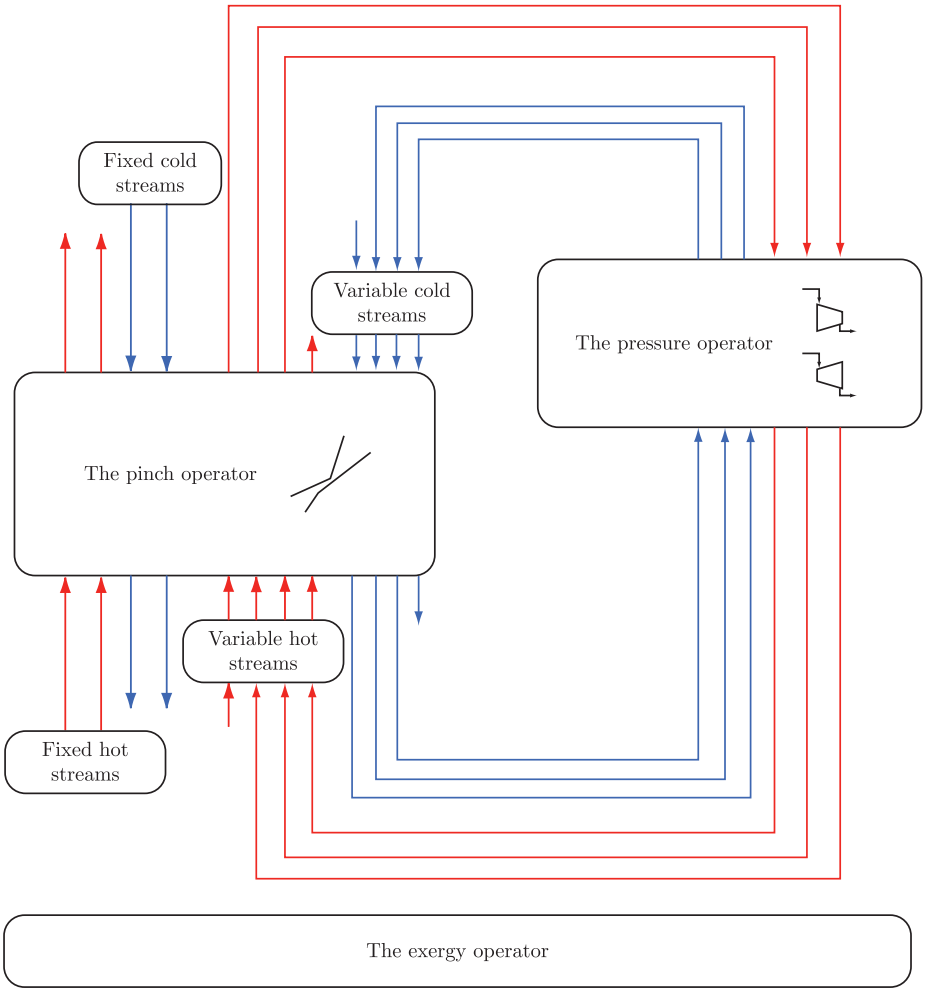


Figure B.2: State space realization of a heat exchanger and compressor/expander network including the exergy operator that transforms energies into exergies to quantify irreversibilities

and compression for variable streams. The former will contribute to the pinch operator as in standard PA, while the latter also interact with the pressure operator. Typically, fixed streams are represented using a constant heat capacity flowrate and the inlet and outlet temperatures. To model phase changes occurring in single component streams, the phenomena can be represented by dividing the stream into 3 substreams: two streams with a constant heat capacity and a third with the latent heat of the phase change at constant temperature. As shown in Figure B.2, the pressure of variable streams is allowed to change through compressors and expanders. This has several implications. First, inlet and outlet temperatures will vary so that the standard transhipment formulation will not be able to solve the problem [55]. Instead, a nonlinear and nonconvex model needs to be applied. Second, one stream can result in up to four contributions to the pinch operator if a maximum of three pressure manipulation stages (compressors and/or expanders) is allowed for each stream. This increases the complexity of the problem considerably as the number of binary variables in the pinch operator scales with the square of the number of streams.

B.3.2 A PA approach for the structure of the HEN and C&E system

In this section, arguments for the most favourable routes for compression and expansion relative to heating and cooling are presented. It will be argued later in this appendix that the appropriate placement of compression and expansion is above and below the pinch, respectively, and with both pressure manipulations preferably starting at the pinch temperature. As a result, an exit stream from a compressor should be cooled to the pinch temperature if expansion (or another compression) is considered as the next step, thus it is a hot stream. Similarly, an exit stream from an expander should be heated to pinch temperature if compression (or another expansion) is considered as the next step, thus it is a cold stream. This is illustrated in Figure 3 that shows the graphical representation of the problem statement for one hot and one cold process stream where a total of three pressure manipulations (e.g., one compressor and two expanders) are allowed for each stream. The hot stream can be cooled, compressed, cooled, expanded, heated, compressed and cooled. Similarly, the cold stream can be heated, expanded, heated, compressed, cooled, expanded and heated.

When supply and target temperatures and pressures of a stream are fixed, there are eight additional variables: three intermediate inlet temperatures, three intermediate outlet temperatures and two intermediate outlet pressures. However, intermediate inlet temperatures, or equivalently exit temperature of the expanders or compressors, are related to the exit pressure of the expanders or compressors. Therefore, there are five independent variables for each stream. The chosen route for compression and expansion of hot and cold streams in Figure B.3 is not arbitrary. The pressure manipulations can be treated as a series of process modifications. In PA, the general approach used to identify process modifications that reduce the energy requirements is called the “plus–minus” principle [110], which states that in general the hot and cold utility targets will be reduced by:

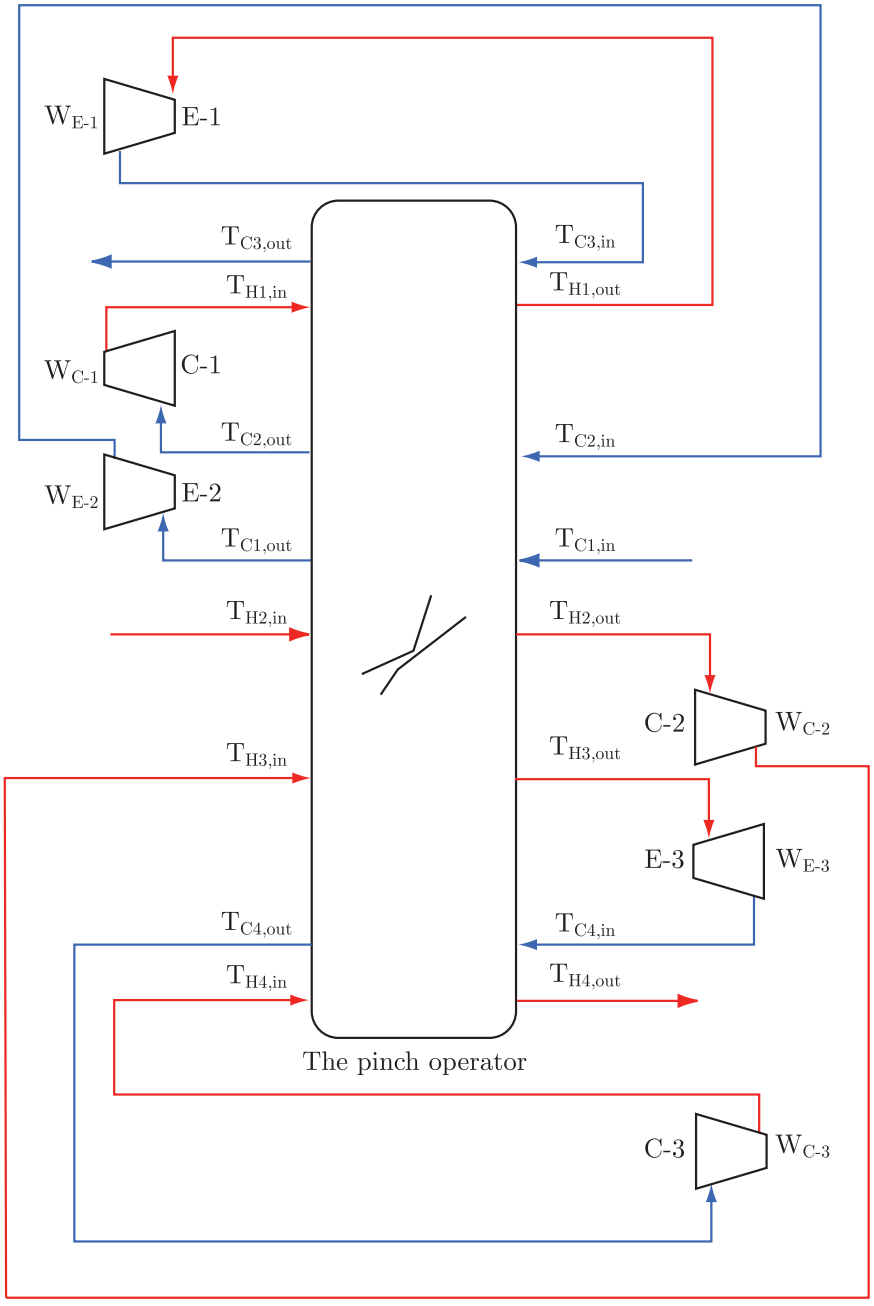


Figure B.3: Superstructure with heat exchangers, compressors and expanders for a hot and a cold stream split into segments showing intermediate temperatures

- increasing hot stream (heat source) duty above the pinch or decreasing hot stream duty below the pinch,
- decreasing cold stream (heat sink) duty above the pinch or increasing cold stream duty below the pinch.

Furthermore, to avoid cross-pinch heat transfer that would result in an increase in consumption of utilities beyond the target, the following applies:

- heat must not be transferred across pinch (from above to below),
- hot utility must not be used below pinch,
- cold utility must not be used above pinch.

One example that applies to the plus–minus principle is a heat pump. A heat pump, if implemented correctly, will transfer heat from below pinch to above pinch, either through an open cycle, such as vapour recompression in distillation, or in a closed heat pump with an external working fluid. In this way, heat is removed below pinch and added above pinch, and will, according to the plus–minus principle, decrease the need for both hot and cold utilities at the expense of the work required in the compressor. Similar to the open cycle heat pump, compression of a gas will increase the temperature of the gas, and thereby either increase the duty of a heat source or decrease the duty of a heat sink. Therefore, by applying the principles above, a stream should always be compressed above pinch temperature. However, compression of a gas at a higher temperature will increase the required work for the same pressure ratio. Although not incorporated here, it should be noted that, from a capital cost point of view, it is beneficial to compress the gas at as low temperature as possible, as the density is higher, and therefore the compressor can be made smaller and less expensive. In some cases, a higher pressure ratio can also be obtained, as the exit temperature will be lower.

An example that demonstrates the principles above is shown in Figure B.4. A hot stream with a heat capacity flow rate of 2 kW/K is to be cooled from 130 °C to -75 °C, and compressed from 0.1 to 0.2 MPa. Two cold streams are to be heated. The first cold stream has a heat capacity flowrate of 5 kW/K and is to be heated from 15 °C to 140 °C. The other cold stream has a heat capacity flowrate of 1 kW/K and is to be heated from -50°C to 140 °C. The hot stream is divided into two segments, H1 and H2, and a compressor is inserted. In addition to the supply and target temperature of the hot stream, two additional intermediate temperatures are introduced: the outlet temperature of segment H1, $T_{H1,\text{out}}$, and the inlet temperature of segment H2, $T_{H2,\text{in}}$. The former corresponds to the temperature at the intake of the compressor while the latter is the exit temperature of the compressor. In this example the compressor intake temperature for the divided hot stream, $T_{H1,\text{out}}$, is varied systematically from the lowest possible temperature (-75°C) to the highest possible temperature (130 °C) in appropriate intervals. In Figure B.4, the composite curves for each of the eight considered cases are shown. In Table B.1, the temperature after compression $T_{H2,\text{in}}$, the work W, the hot and cold utilities, Q_H and Q_C, respectively,

Case	$T_{H1,\text{out}}$ [°C]	$T_{H2,\text{in}}$ [°C]	Q_H [kW]	Q_C [kW]	W [kW]	ψ [%]
1	—	—	540.0	135.0	—	56.3
2	−75.0	−31.6	540.0	221.7	86.7	66.4
3	−50.0	−1.2	540.0	232.7	97.7	63.1
4	−28.5	25.0	540.0	242.0	107.1	61.1
5	0.0	59.8	470.4	185.0	119.6	67.0
6	25.0	90.3	409.5	135.0	130.5	73.2
7	50.0	120.7	398.5	135.0	141.5	71.8
8	130.0	218.3	363.5	135.0	175.5	67.6

Table B.1: Effect of compression of a hot stream at varying compressor intake temperatures on utility requirements and exergy efficiency

and the exergy efficiency ψ, which will be formally defined in Section B.4.3, are given. Figure B.5 shows the variation of W, Q_H, Q_C, and ψ as a function of compressor intake temperature $T_{H1,\text{out}}$. In the calculations, a minimum temperature approach $\Delta T_{\min} = 10°\text{C}$ and isentropic compression of an ideal gas with a polytropic exponent of $\kappa = 1.4$ are assumed.

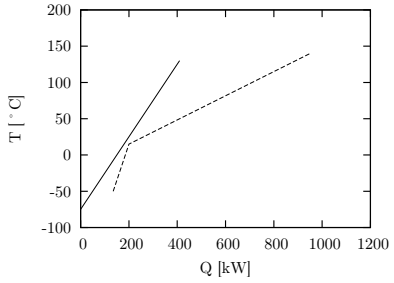
The first case (Figure B.4(a)) shows the composite curves (CCs) for the three streams without compression. As can be seen from Table B.1, the hot and cold utilities are 540 and 135 kW, respectively, and the exergy efficiency is 56.3% for a hot stream pressure of 0.1 MPa (pressure-based exergy is not included). Since the heat capacity flow rate of the CC for the cold streams is always larger than the CC for the hot streams above pinch, the pinch temperature (15 °C/25 °C) does not change throughout the example.

In the second case, the hot stream is compressed at the lowest possible temperature. As can be seen from Table B.1, the required work is only 86.7 kW, however, since it is compressed solely below the pinch point, it leads to an increase in cold utility, which is in accordance with the plus–minus principle. In contrast to the previous case, the pressure exergy is utilized so that exergy efficiency is increased to 66.4%.

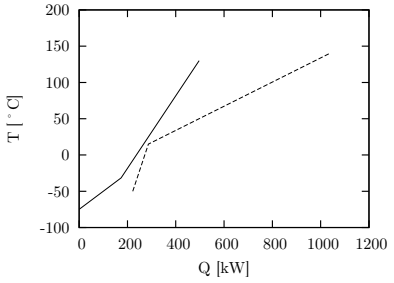
In Case 3, the compression temperature is increased to -50 °C. As a result, the work is increased to 97.7 kW and since the compressor exit temperature is below the pinch, there is an equal increase in cold utility. It can therefore be concluded that if a stream has to be compressed below the pinch, it should be compressed at as low temperature as possible.

In Case 4, the exit temperature (25 °C) is exactly equal to the hot pinch temperature resulting in the worst possible placement of the compressor and an exergy efficiency of only 61.1%. This can be explained by the maximum amount of heat resulting from the compression delivered below the pinch. It increases the cold utility without reducing the hot utility, with an increase in work from Case 3.

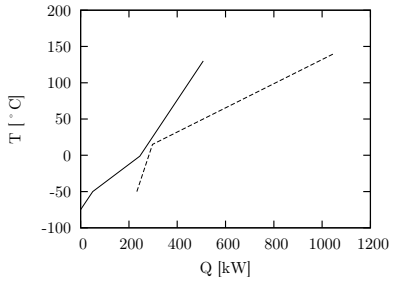
In Case 5, the compression is performed across the pinch (from below to above) and the temperatures before and after compression are 0.0 °C and 59.8 °C, respectively, which means that a portion of the heat due to compression is provided above pinch. This will reduce the hot and cold utilities duties, as the plus–minus principle states, and increase



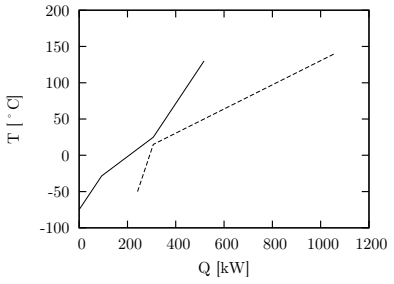
(a) Case 1: No Compression



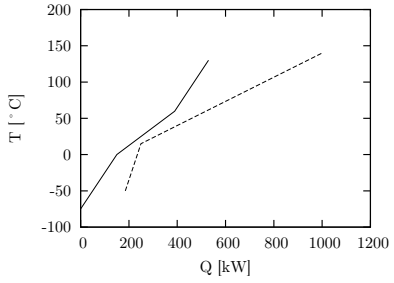
(b) Case 2: Compression at $T_{H1,\text{out}} = -75\,^{\circ}\text{C}$



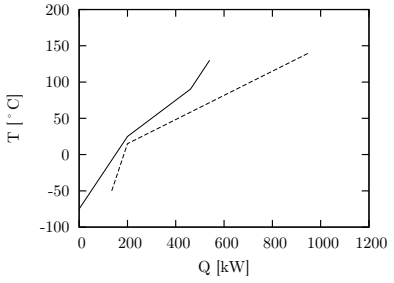
(c) Case 3: Compression at $T_{H1,\text{out}} = -50\,^{\circ}\text{C}$



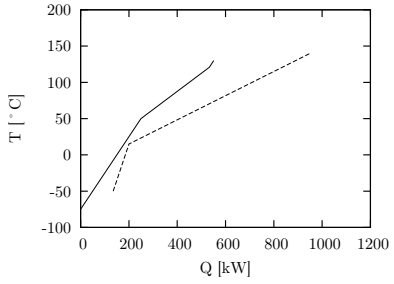
(d) Case 4: Compression at $T_{H1,\text{out}} = -28.5\,^{\circ}\text{C}$



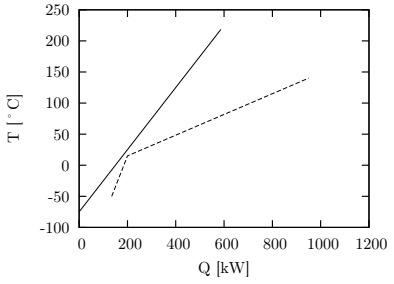
(e) Case 5: Compression at $T_{H1,\text{out}} = 0\,^{\circ}\text{C}$



(f) Case 6: Compression at $T_{H1,\text{out}} = 25\,^{\circ}\text{C}$



(g) Case 7: Compression at $T_{H1,\text{out}} = 50\,^{\circ}\text{C}$



(h) Case 8: Compression at $T_{H1,\text{out}} = 130\,^{\circ}\text{C}$

Figure B.4: Composite Curves resulting from compression of a hot stream at varying compressor intake temperatures

the exergy efficiency when compared to Case 4.

Case 6 is the optimal configuration with an exergy efficiency of 73.2%. Here, the hot stream is cooled to hot pinch temperature (25 °C) before it is compressed to 0.2 MPa and 90.3°C. Although the work has increased to 130.5 kW, all the heat from the compressor is now provided above pinch and will therefore reduce the hot and cold utilities duty to 409.5 and 135 kW, respectively. As can be seen, the cold utility is now the same as in Case 1 where no compression took place. Furthermore, note that the CCs are close over a large interval, which is an indication of the small irreversibilities in the HEN.

Increasing the compressor intake temperature further requires additional work, which will be recovered as heat. However, since work is always worth more than heat above ambient temperature, this process leads to a degradation of exergy. Therefore, the exergy efficiency continues to decrease in Case 7 and is at its lowest value for compression above pinch in Case 8. It could be argued that in the last case the hot utility could be provided at a lower temperature since the temperature after compression is higher than the cold stream outlet temperature plus the minimum internal temperature approach of 10 °C. However, even when accounting for this effect, the exergy efficiency will still be lower than for Case 6.

As demonstrated in Figure B.5, it is best to compress the hot stream beginning from the pinch point than from any temperature below the pinch point if one strives for high exergy efficiency. Cases 2 through 5 violate the plus–minus principle, they increase the hot stream duty below the pinch. On the other hand, Cases 7 and 8 lead to a degradation of exergy, that is, conversion from work to heat.

As already mentioned, for this example the heat capacity flowrate for the cold CC above the pinch is larger than the heat capacity flowrate for the hot stream. Hence, the pinch point remains the same throughout the example. In the opposite situation, the pinch point will coincide with the compressor exit temperature and actually increase as the intake temperature to the compressor is increased, thereby reducing the benefit of compression above the pinch point. In addition, since in most problems several streams are involved, the pinch point is likely to "jump" from one location to another when the pressure of process streams is manipulated.

A similar analysis can be performed for the case of expansion. It will decrease the temperature of the process stream, and thereby either decrease the duty of a heat source or increase the duty of a heat sink. Applying the plus–minus principle, a stream should always be expanded to an expander exit temperature that is below the pinch. This is certainly the fact for all refrigeration cycles. However, similar to compression, the expansion work will be greater at higher temperatures. Furthermore, a larger cold duty will be produced. Therefore, it is obvious that expansion of a gas should start at the pinch temperature and end below the pinch temperature. If the sole purpose of the expansion is to produce work, it is favourable to expand the gas at as high temperature as possible, which is clearly the case for power production plants. Nevertheless, in many processes where heat integration is important, and especially for processes that require refrigeration, generation of work is secondary to providing heat sinks. An example and a discussion of expansion of a cold stream below the pinch can be found in [9].

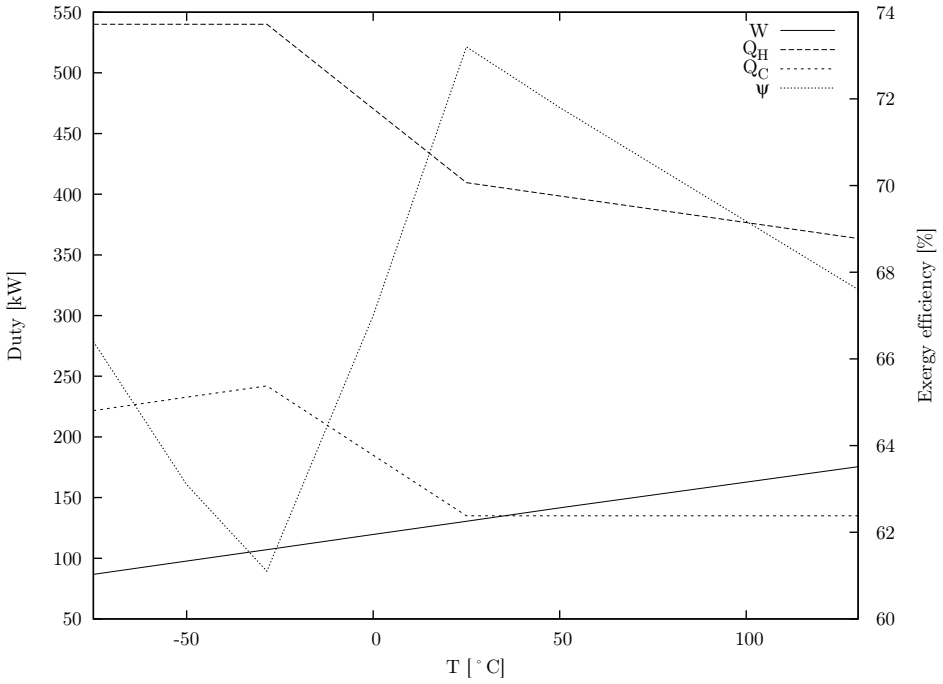


Figure B.5: Exergy efficiency, required work and utilities for the example

In general, a stream without phase change and with a supply pressure equal to the target pressure should not be compressed or expanded. However, in some cases it may be beneficial to expand, compress and expand (or vice versa) the same stream with intermediate cooling and heating to take advantage of heat pockets created by other process streams at higher or lower temperatures. To do so work must be available, of course. One example of such a refrigeration cycle is the inverse Brayton cycle (expander cycle) commonly used in air separation and peak-shave LNG plants.

For a liquid stream, there is no reason for manipulating the pressure as the effect is marginal. However, for a stream with phase change, pumping in the liquid phase and expansion in the gas phase is a very interesting option that should be investigated. A more thorough discussion with heuristics for how to utilize pressure manipulations in process design can be found in [9].

From the previous discussion, it can be concluded that the most beneficial way of manipulating the pressure of a hot stream is to cool, compress, cool, expand, heat, compress and cool it. Furthermore, the stream should always be compressed to temperatures above the pinch point and expanded to temperatures below the pinch point. Compression and expansion of a hot stream should preferably start at the hot pinch temperature. Therefore, as shown in Figure B.3, an expanded stream will always be a cold stream, whereas a compressed stream will always be a hot stream when further pressure changes are to be made. Similarly, a cold stream can be heated, expanded, heated, compressed, cooled, expanded and heated. Compression and expansion of a cold stream should preferably

start at the cold pinch temperature. Note that during this process, a hot stream may temporarily change to be a cold stream, and a cold stream may temporarily change to be a hot stream. It is also worth noticing that the location of the pinch point is very likely to change once pressure manipulations are introduced, since the shape of the composite curve will change. This suggests the use of an optimization model to find the best trade-off as even small problems become intractable.

B.4 Model formulation

The model consists of four parts: the pinch operator, the pressure operator, the exergy operator and the objective function, recall Figure B.2. The pinch operator is responsible for locating the pinch point and subsequently determining the minimum required utilities. In the case of constant heat capacity flowrates, the pinch operator is linear. The pressure operator uses equations of state in combination with isentropic changes of state to connect streams at different pressure levels. With the thermodynamic model, nonconvex constraints are introduced that cannot be reformulated. Typically, objective functions corresponding to variable costs are linear. Together these operators form an MINLP involving nonconvex functions.

B.4.1 The Pinch Operator

At the heart of the optimization model is the pinch operator, which calculates the minimum required hot and cold utilities. Since stream pressure is allowed to change through compressors and expanders, the inlet temperatures to the pinch operator will vary, thus creating difficulties for the temperature interval model as first proposed by Linnhoff and Flower [108] and modeled as a transhipment problem in [132]. Restructuring the temperature intervals implies making discrete changes that lead to non-differentiabilities in the model.

Duran and Grossmann proposed an optimization formulation to find the minimum utility requirement for cases with variable stream data [55]. The pinch location method was developed to optimize and heat integrate chemical processes simultaneously, and therefore allows for variable inlet temperatures and heat capacity flowrates to the pinch operator. The presented formulation locates the pinch point by comparing the hot utility required for the subsystem of streams above each pinch candidate. It yields a small model without binary variables, but uses nonsmooth functions. Thus, the problem cannot be solved with standard optimization algorithms and smooth approximations were introduced. A reformulation of this model based on disjunctive programming was presented in [71]. This reformulation removes the nonsmooth constraint at the expense of binary variables, which distinguish if a stream is always below a candidate pinch point, above it or crosses it. A candidate pinch point can be any stream inlet temperature. Hence, each hot or cold stream is regarded as a possible pinch candidate. This model is a MILP if heat capacity flowrates are non-varying; otherwise, it leads to a MINLP.

Another pinch operator that allows variable temperatures was presented in a series of three papers in [179–181]. Here, a superstructure is proposed that matches each hot and cold stream at different stages. It uses binary variables to assign a possible heat exchange between hot and cold streams in addition to heat balances and temperature constraints. Although more complex than the pinch point location model, this pinch operator has the advantage of calculating the required area as part of the optimization routine. Unfortunately, this leads to a very large number of binary variables and the model can therefore not be used for the considered purposes.

Recently, the model by Yee et al. was extend by Ponce-Ortega et al. [134] to include isothermal process streams and by Hasan et al. [77] for process streams with nonconstant heat capacity. In the latter case, a significant number of binary variables is added to the model rendering it too complicated for this application.

It has been found in this present research that the formulation given by Grossmann et al. [71] is the most tractable MINLP formulation currently available for the pinch operator. Their model can also include the case of isothermal streams, which is neglected in the presentation here. As in the original paper, the big M formulation is used. For completeness, it can be described as follows.

Given is a set of hot process streams H and a set of cold process streams C. Let $S = H \cup C$ be the set of all process streams. For each $s \in S$, $T_{s,\text{in}}$, $T_{s,\text{out}}$, F_s, and $c_{p,s}$ denote the inlet and outlet temperature, the flowrate and the heat capacity of the stream, respectively. Let $k \in H$ and $l \in C$ be indices of pinch candidates. Let $QHOT_i$ be the energy to be transferred from hot stream i, $QCOLD_j$ the energy to be transferred to cold stream j. Let q_{ki}^{hp} and q_{li}^{cp} be the energy available from hot stream i above hot pinch candidate k and cold pinch candidate l, respectively. Likewise, let q_{kj}^{hp} and q_{lj}^{cp} be the energy required by cold stream j above pinch candidate k and l, respectively. The hot and cold utilities used are Q_H and Q_C. The minimum approach temperature between hot and cold streams is given by $\Delta T_{\min}$. Lastly, binary variables w_{ki}^1, w_{ki}^2 and w_{ki}^3 denote if hot stream i is completely above, crosses or is completely below hot pinch candidate k, respectively. z_{kj}^1, z_{kj}^2, z_{kj}^3, u_{li}^1, u_{li}^2, u_{li}^3, v_{lj}^1, v_{lj}^2 and v_{lj}^3 denote analogous cases for the other combinations of streams and pinch candidates as indicated by the indices. U and M are upper bounds on the heat transfer and temperatures, ε is a small parameter introduced to distinguish numerically if a stream crosses the pinch candidate or is below the pinch candidate.

$$
\begin{aligned}
Q_H + \sum_{i \in H} QHOT_i &= Q_C + \sum_{j \in C} QCOLD_j, \\
Q_H &\geq \sum_{j \in C} q_{kj}^{hp} - \sum_{i \in H} q_{ki}^{hp}, \quad \forall k \in H, \\
Q_H &\geq \sum_{j \in C} q_{lj}^{cp} - \sum_{i \in H} q_{li}^{cp}, \quad \forall l \in C, \\
QHOT_i &= F_i c_{p,i} (T_{i,in} - T_{i,out}), \quad \forall i \in H,
\end{aligned}
$$

$$
\begin{aligned}
QCOLD_j &= F_j c_{p,j}(T_{j,out} - T_{j,in}), \quad \forall j \in C, \\
q_{ki}^{hp} - QHOT_i &\leq U(1 - w_{ki}^1), \quad \forall (i,k) \in H \times H, \\
T_{i,in} &\geq T_{k,in} - M(1 - w_{ki}^1), \quad \forall (i,k) \in H \times H, \\
T_{i,out} &\geq T_{k,in} - M(1 - w_{ki}^1), \quad \forall (i,k) \in H \times H, \\
q_{ki}^{hp} - F_i c_{p,i}(T_{i,in} - T_{k,in}) &\leq U(1 - w_{ki}^2), \quad \forall (i,k) \in H \times H, \\
T_{i,in} &\geq T_{k,in} - M(1 - w_{ki}^2), \quad \forall (i,k) \in H \times H, \\
T_{i,out} &\leq T_{k,in} - \varepsilon + M(1 - w_{ki}^2), \quad \forall (i,k) \in H \times H, \\
q_{ki}^{hp} &\leq U(1 - w_{ki}^3), \quad \forall (i,k) \in H \times H, \\
T_{i,in} &\leq T_{k,in} - \varepsilon + M(1 - w_{ki}^3), \quad \forall (i,k) \in H \times H, \\
T_{i,out} &\leq T_{k,in} - \varepsilon + M(1 - w_{ki}^3), \quad \forall (i,k) \in H \times H, \\
w_{ki}^1 + w_{ki}^2 + w_{ki}^3 &= 1, \quad \forall (i,k) \in H \times H, \\
q_{kj}^{hp} - QCOLD_j &\geq -U(1 - z_{kj}^1), \quad \forall (j,k) \in C \times H, \\
T_{j,in} &\geq T_{k,in} - \Delta T_{\min} - M(1 - z_{kj}^1), \quad \forall (j,k) \in C \times H, \\
T_{j,out} &\geq T_{k,in} - \Delta T_{\min} - M(1 - z_{kj}^1), \quad \forall (j,k) \in C \times H, \\
q_{kj}^{hp} - F_j c_{p,j}(T_{j,out} - (T_{k,in} - \Delta T_{\min})) &\geq -U(1 - z_{kj}^2), \quad \forall (j,k) \in C \times H, \\
T_{j,in} &\leq T_{k,in} - \Delta T_{\min} + M(1 - z_{kj}^2), \quad \forall (j,k) \in C \times H, \\
T_{j,out} &\geq T_{k,in} - \Delta T_{\min} - \varepsilon - M(1 - z_{kj}^2), \quad \forall (j,k) \in C \times H, \\
q_{kj}^{hp} &\leq U(1 - z_{kj}^3), \quad \forall (j,k) \in C \times H, \\
T_{j,in} &\leq T_{k,in} - \Delta T_{\min} - \varepsilon + M(1 - z_{kj}^3), \quad \forall (j,k) \in C \times H, \\
T_{j,out} &\leq T_{k,in} - \Delta T_{\min} - \varepsilon + M(1 - z_{kj}^3), \quad \forall (j,k) \in C \times H, \\
z_{kj}^1 + z_{kj}^2 + z_{kj}^3 &= 1, \quad \forall (j,k) \in C \times H, \\
q_{li}^{cp} - QHOT_i &\leq U(1 - u_{li}^1), \quad \forall (i,l) \in H \times C, \\
T_{i,in} &\geq T_{l,in} + \Delta T_{\min} - M(1 - u_{li}^1), \quad \forall (i,l) \in H \times C, \\
T_{i,out} &\geq T_{l,in} + \Delta T_{\min} - M(1 - u_{li}^1), \quad \forall (i,l) \in H \times C, \\
q_{li}^{cp} - F_i c_{p,i}(T_{i,in} - (T_{l,in} + \Delta T_{\min})) &\leq U(1 - u_{li}^2), \quad \forall (i,l) \in H \times C, \\
T_{i,in} &\geq T_{l,in} + \Delta T_{\min} - M(1 - u_{li}^2), \quad \forall (i,l) \in H \times C, \\
T_{i,out} &\leq T_{l,in} + \Delta T_{\min} - \varepsilon + M(1 - u_{li}^2), \quad \forall (i,l) \in H \times C, \\
q_{li}^{cp} &\leq U(1 - u_{li}^3), \quad \forall (i,l) \in H \times C, \\
T_{i,in} &\leq T_{l,in} + \Delta T_{\min} - \varepsilon + M(1 - u_{li}^3), \quad \forall (i,l) \in H \times C, \\
T_{i,out} &\leq T_{l,in} + \Delta T_{\min} - \varepsilon + M(1 - u_{li}^3), \quad \forall (i,l) \in H \times C, \\
u_{li}^1 + u_{li}^2 + u_{li}^3 &= 1, \quad \forall (i,l) \in H \times C,
\end{aligned}
$$

$$
\begin{aligned}
q_{lj}^{cp} - QCOLD_j &\geq -U(1 - v_{lj}^1), \quad \forall (j,l) \in C \times C, \\
T_{j,in} &\geq T_{l,in} - M(1 - v_{lj}^1), \quad \forall (j,l) \in C \times C, \\
T_{j,out} &\geq T_{l,in} - M(1 - v_{lj}^1), \quad \forall (j,l) \in C \times C, \\
q_{lj}^{cp} - F_j c_{p,j}(T_{j,out} - T_{l,in}) &\geq -U(1 - v_{lj}^2), \quad \forall (j,l) \in C \times C, \\
T_{j,in} &\leq T_{l,in} + M(1 - v_{lj}^2), \quad \forall (j,l) \in C \times C, \\
T_{j,out} &\geq T_{l,in} - \varepsilon - M(1 - v_{lj}^2), \quad \forall (j,l) \in C \times C, \\
q_{lj}^{cp} &\leq U(1 - v_{lj}^3), \quad \forall (j,l) \in C \times C, \\
T_{j,in} &\leq T_{l,in} - \varepsilon + M(1 - v_{lj}^3), \quad \forall (j,l) \in C \times C, \\
T_{j,out} &\leq T_{l,in} - \varepsilon + M(1 - v_{lj}^3), \quad \forall (j,l) \in C \times C, \\
v_{lj}^1 + v_{lj}^2 + v_{lj}^3 &= 1, \quad \forall (j,l) \in C \times C.
\end{aligned}
$$

B.4.2 The Pressure Operator

Let p_s be the pressure of stream s where $s \in S$. The pair $(s_1, s_2) \in EX \subset (S \times C)$ denotes that the outlet of stream s_1 from the pinch operator is connected to the inlet of stream s_2 to the pinch operator with an expander. Likewise, the pair $(s_1, s_2) \in CO \subset (S \times H)$ denotes connection with a compressor. Note that in- and outlet of the streams refer to the heat exchanger, not the compressor or expander. $\tilde{T}_{s_2}$ denotes the exit temperature of a reversible process, κ is the polytropic exponent, η_C and η_E are the isentropic efficiencies of the compressors and expanders, respectively. W_{s_1} denotes the work required or released by compression or expansion of stream s_1. The reversible and adiabatic compression or expansion of an ideal gas can be formulated as follows.

$$
\begin{aligned}
(\kappa - 1)\ln p_{s_1} + \kappa \ln \tilde{T}_{s_2} &= (\kappa - 1)\ln p_{s_2} + \kappa \ln T_{s_1}, \quad \forall (s_1, s_2) \in CO \cup EX, \\
\left(T_{s_1} - \tilde{T}_{s_2}\right) &= \left(T_{s_1} - T_{s_2}\right)\eta_C, \quad \forall (s_1, s_2) \in CO, \\
\left(T_{s_1} - \tilde{T}_{s_2}\right)\eta_E &= \left(T_{s_1} - T_{s_2}\right), \quad \forall (s_1, s_2) \in EX, \\
W_{s_1} &= F_{s_1} c_{p,s_1}\left(T_{s_2} - T_{s_1}\right), \quad \forall (s_1, s_2) \in CO, \\
W_{s_1} &= F_{s_1} c_{p,s_1}\left(T_{s_1} - T_{s_2}\right), \quad \forall (s_1, s_2) \in EX.
\end{aligned}
\tag{B.1}
$$

The logarithmic terms in Eq. (B.1) involve positive, physical quantities for which tighter bounds are established to avoid the logarithm becoming undefined. Note that compressor and expander work are both defined as nonnegative quantities. Hence, the net work produced equals the sum of the expansion work minus the sum of the compression work. In the general case when the thermodynamic properties of the streams are described with a volume-explicit equation of state, the equations are to be rewritten accordingly. A detailed discussion on how to compute the unknown exit temperature and the work for

an isentropic process can be found in Prausnitz [135]. Lastly, it should be noted that in the case studies no pressure drop is considered in the heat exchangers, although a constant pressure drop can be easily considered in the given model.

B.4.3 The Exergy Operator

The exergy operator has two purposes: to calculate the exergy of the process streams and the utilities and to find the exergy conversion efficiency. According to Kotas [101], work is defined as 100% exergy, whereas the exergy of the hot and cold utilities depends on the ambient temperature as well as the utility temperature and duty. For a utility with constant temperature, the Carnot efficiency can be used to find the exergy content. Note that we have assumed the hot utility to be above the ambient temperature and the cold utility to be below ambient temperature. Let T_0 and p_0 be ambient temperature and pressure, respectively. T_U^h and T_U^c are the temperatures at which hot and cold utilities are provided. One can interpret the exergy operator as a pricing tool to derive cost coefficients for the different utilities thermodynamically so that the optimal solution corresponds to minimum irreversibility.

$$
\begin{aligned}
ExW &= \sum_{(s_1,s_2)\in CO} W_{s_1} - \sum_{(s_1,s_2)\in EX} W_{s_1} \\
ExQhu &= Q_H \left(1 - \frac{T_0}{T_U^h}\right) \\
ExQcu &= Q_C \left(\frac{T_0}{T_U^c} - 1\right)
\end{aligned}
$$

The thermo-mechanical exergy of the process streams consists of the temperature-based exergy and the pressure-based exergy. Since we assume that the process streams are non-isothermal, a logarithmic expression must be used to calculate the exergy content as shown in Equation (B.2). The pressure exergy is defined by Equation (B.3). The expressions can be derived using the first and second laws of thermodynamics and the ideal gas model with constant heat capacities. A derivation of the exergy expressions can be found in [101].

$$
E_s^{(T)} = F_s c_{p,s} \left[T_s - T_0 \left(1 + \ln\left(\frac{T_s}{T_0}\right)\right)\right] \tag{B.2}
$$

$$
E_s^{(p)} = F_s T_0 R \ln\left(\frac{p_s}{p_0}\right) = F_s \left(\frac{\kappa - 1}{\kappa}\right) c_{p,s} T_0 \ln\left(\frac{p_s}{p_0}\right) \tag{B.3}
$$

When calculating the exergy conversion efficiency it is important to exclude contributions that do not change during the process in order to get a representative measure. The chemical exergy is therefore excluded from the calculations and only the thermo-mechanical exergy is included.

$$E_s^{(tm)} = E_s^{(T)} + E_s^{(p)}$$

A more thorough discussion about thermo-mechanical exergy for subambient process streams can be found in [9]. The exergy conversion efficiency is defined as the useful outlet exergy divided by the inlet exergy. The inlet exergy is defined as the sum of the thermo-mechanical exergy of inlet process streams and utilities and the net work required, whereas the outlet useful exergy is the sum of the thermo-mechanical exergy in the outlet streams and the net work produced as stated in Equation (B.4).

$$\psi = \frac{E^{(tm)}_{\text{outlet streams}} + \sum_{(s_1,s_2)\in EX} W_{s_1}}{E^{(tm)}_{\text{inlet streams}} + \sum_{(s_1,s_2)\in CO} W_{s_1} + ExQcu + ExQhu} \tag{B.4}$$

Streams with fixed temperatures contribute only temperature-based exergy as the pressure remains constant throughout the process, whereas the streams with variable temperature include both the pressure- and temperature-based exergy, that is, the total thermo-mechanical exergy. The exergy input from the utilities is included in Equation (B.4). The net required work and net generated work are also included in the exergy efficiency. If the heat capacity flowrates, supply and target temperatures and pressures of the process streams are fixed, the highest exergy efficiency can be found by minimizing the exergy input of the required work, hot and cold utilities.

B.4.4 The Objective Function

The objective function combines the results from pinch, pressure and exergy operators into one measure. For example, when the stream heat capacity flowrates and the inlet and outlet pressure and temperatures are constant, the exergy required by the design can be minimized resulting in a solution with minimal irreversibilities.

$$\min \; ExW + ExQhu + ExQcu \tag{B.5}$$

If flowrates are allowed to vary, more care needs to be taken in order to ensure that the exergy of the streams with variable flowrate is accounted for.

Alternatively, it is possible to assign different costs for the utilities and work than those obtained from thermodynamical considerations.

B.5 Examples

All problems were solved in GAMS 23.2 using BARON with CPLEX and SNOPT on a Intel Xeon W3570 workstation using one core at 3.20GHz and 4 GB RAM under Linux 2.6.28. The relative termination tolerance in GAMS, `OptCR`, was set to 10^{-4}, while the absolute termination tolerance in GAMS, `OptCA`, was not changed and the default value given by BARON, 10^{-9}, was used. No deviating tolerances were set for either SNOPT

Stream	$F_s c_{p,s}$[kW/K]	$T_{s,\text{in}}$ [K]	$T_{s,\text{out}}$ [K]	p_s [MPa]
H1	3	288	123	0.1
C1	2	213	288	0.1
C2	1.7	113	—	0.4
C3	1.7	—	—	—
H2	1.7	—	—	—
C4	1.7	—	288	0.1

Table B.2: Given information for stream in simple example

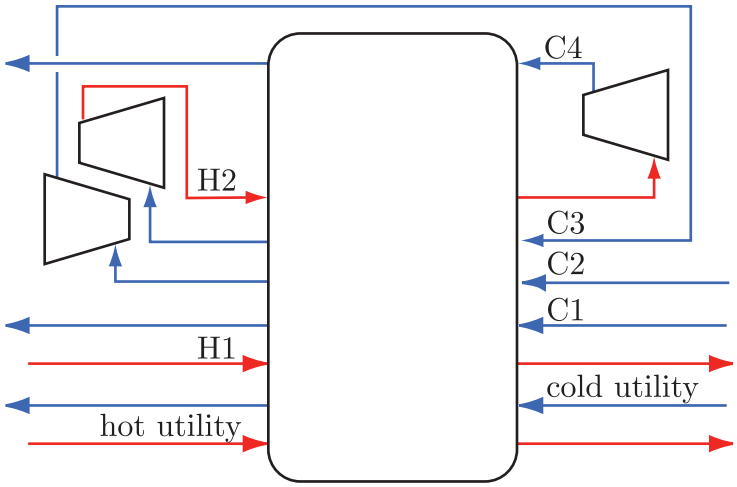


Figure B.6: Possible arrangement of streams in the simple example

or CPLEX. Since BARON is used, bounds on the variables need to be given. Bounds on temperatures are given by the available utility temperatures, bounds on pressures are specified individually in the examples.

B.5.1 A simple example

In this example the benefits of using the model formulation presented in this appendix are illustrated. One hot stream (H1) and one cold stream (C1) are at constant pressure, a second cold stream (C2) is to be expanded from 0.4 MPa to 0.1 MPa. In light of the discussion earlier, C2 could potentially be expanded, compressed and expanded with necessary cooling and/or heating. In this example, the heat capacity flowrates of all streams are constant. The data for each stream are given in Table B.2, the connection and labelling of streams are shown in Figure B.6.

Furthermore, $\Delta T_{\min} = 4\,\text{K}$, $T_0 = 288\,\text{K}$, $p_0 = 0.1\,\text{MPa}$, $T_U^h = 383\,\text{K}$, $T_U^c = 93\,\text{K}$, $\kappa = 1.352$ and $\eta_C = \eta_E = 1$. Unknown inlet temperatures can be varied between 103 to 373 K, the pressure of stream C3 is restricted to 0.1–0.4 MPa, the pressure of H2 to 0.1–0.6 MPa.

Stream	$T_{s,\text{in}}$ [K]	$T_{s,\text{out}}$ [K]	p_s [MPa]
C2	—	155.56	—
C3	126.85	201.56	0.183
H2	244.21	233.00	0.382
C4	164.39	—	—

Table B.3: Result for decision variables for Case 2 of the simple example

Several different cases are studied. First, no expanders and compressors are used. Then, to contrast the possible benefits of adjusting pressure levels of intermediate streams, three additional cases are presented where the objective and some constraints are modified.

The CCs and GCC for the base case without pressure manipulation are shown in Case 1 in Figure B.7. No work is produced and the heating and cooling utilities are 64.5 and 112 kW, respectively, giving a thermo-mechanical exergy efficiency of 68.1%. The change in pressure exergy for stream C2 is ignored in these initial calculations. If a valve is used and the pressure change in C2 is accounted for, the exergy efficiency will be as low as 39.2%. As can be seen from the GCC, the pinch point is at 217 K/213 K.

In Case 2, the model formulation is used to find the minimum irreversibilities given the possible path from Figure B.6. This is formulated using the objective function given by Equation (B.5). It is found to be optimal that stream C2 is expanded to 0.183 MPa, recompressed to 0.382 MPa and finally expanded down to 0.1 MPa. The net work produced by the process is 92.96 kW, the hot utility requirement is reduced to 45.46 kW and no cold utility is necessary. The resulting exergy efficiency is 91.4% if one assumes that the net work produced can be utilized elsewhere. Results for the intermediate state variables are listed in Table B.3. The problem was solved in 4 hours and 42 minutes. There are four pinch points, at 130.85 K/126.85 K, at 168.37 K/164.37 K, at 217 K/213 K and at 244.21 K/240.21 K. Note that the GCC seems to indicate an additional pinch point at 117 K/113 K. However, this is a result of the fact that the designed process requires no cold utility and, therefore, it does not indicate the existence of an additional pinch point. This large number of pinch points corresponds to the objective of minimizing irreversibilities and therefore decreasing the gap between hot and cold composite curves as far as possible. It is worth mentioning that this configuration and the most favorable intermediate temperature for expanding this stream could also have been found by the ExPAnD methodology [9]. The advantage of the optimization approach of this appendix is time saving and assurance of optimality. It is also worth noticing that the result from Case 2 is in agreement with the proposed model in Figure B.6. At first sight, this example seems innocuous. However, it is indeed a difficult global optimization problem as the ideal gas model introduces nonconvexities and the pinch operator introduces 108 binary variables. Furthermore, the existence of multiple pinch points at the optimal solution introduces degeneracy that slows the solution algorithm significantly.

In Case 3, the system is evaluated on an energy basis where hot and cold utility duties as well as the work provided to the process are minimized. The objective function is given

Stream	$T_{s,\text{in}}$ [K]	$T_{s,\text{out}}$ [K]	p_s [MPa]
C2	—	197.18	—
C3	142.71	263.99	0.116
H2	304.76	123.47	0.201
C4	103.00	—	—

Table B.4: Result for decision variables for Case 3 of the simple example

Stream	$T_{s,\text{in}}$ [K]	$T_{s,\text{out}}$ [K]	p_s [MPa]
C2	—	180.07	—
C3	125.52	251.65	0.100
H2	296.71	121.44	0.188
C4	103.00	—	—

Table B.5: Result for decision variables for Case 4 of the simple example

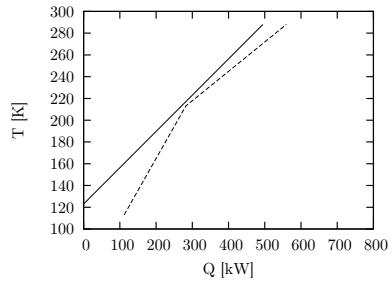
by

$$\min W + Q_H + Q_C. \tag{B.6}$$

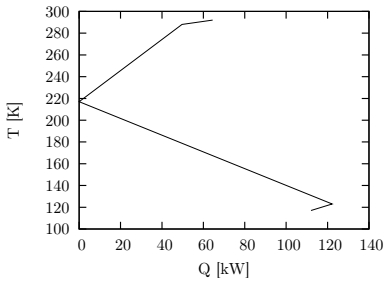
Here, it is optimal to expand stream C2 to 0.116 MPa, recompress to 0.201 MPa and then expand to 0.1 MPa. Results for the intermediate state variables are listed in Table B.4. The total net work produced is 58.08 kW and the cold and hot utilities are 0 kW and 10.58 kW, respectively, giving an exergy efficiency of 84.9%. The problem was solved in 7 seconds. In this case, there are two pinch points, one at 217 K/213 K and one at 288 K/284 K.

In Case 4, the minimal irreversibilities are found while fixing the hot and cold utilities to zero. In this case, minimizing irreversibilities is the same as maximizing net work produced. Stream C2 is expanded from 0.4 MPa to 0.1 MPa, recompressed to 0.188 MPa and expanded to 0.1 MPa. Results for the intermediate temperatures are listed in Table B.5. The total net work produced is 47.5 kW and the hot and cold utilities are both 0 kW, giving an exergy efficiency of 83.2%. In this case, the problem is completely balanced by the hot and cold streams and sub-streams so that there is no need for hot or cold utilities. The problem was solved in less than 2 seconds. There exists one pinch point at 288 K/284 K and a near pinch at 217 K/213 K. Again, the GCC touches the temperature axis two additional times at both ends which indicates that neither cold nor hot utility is required.

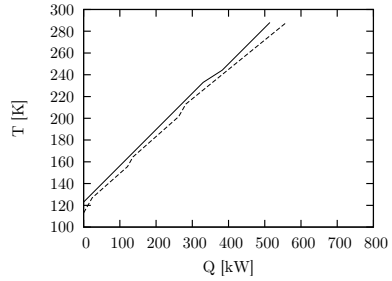
By adding additional passes of C2 through the heat exchanger, expanders and compressors, the exergy efficiency can be further increased; however, the investment cost would be increased significantly while the benefits diminish. It is also possible to set the isentropic efficiencies for the compressors and expanders to a value less than 100%. This will reduce the exergy efficiency, but lead to a more realistic process model. Furthermore, the model allows for changes in the heat capacity flowrate of the process streams. In this way, constraints on the utilities as well as the net produced work can be set, forcing them to be zero. This is done in the next example, where an LNG process, which is self-supporting



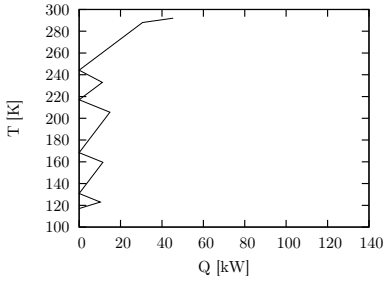
(a) Composite Curves for Case 1



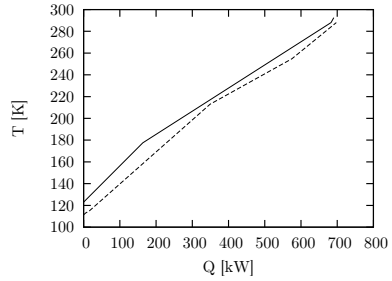
(b) Grand Composite Curve for Case 1



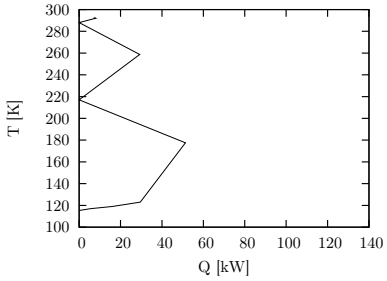
(c) Composite Curves for Case 2



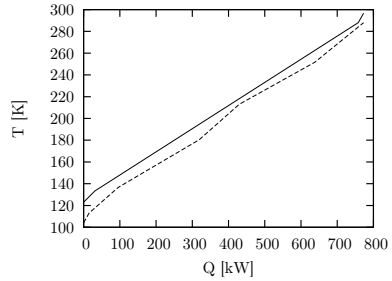
(d) Grand Composite Curve for Case 2



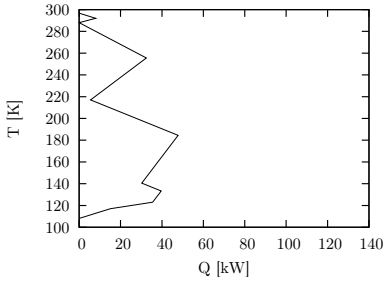
(e) Composite Curves for Case 3



(f) Grand Composite Curve for Case 3



(g) Composite Curves for Case 4



(h) Grand Composite Curve for Case 4

Figure B.7: Composite and Grand Composite Curves for the different cases in the simple example

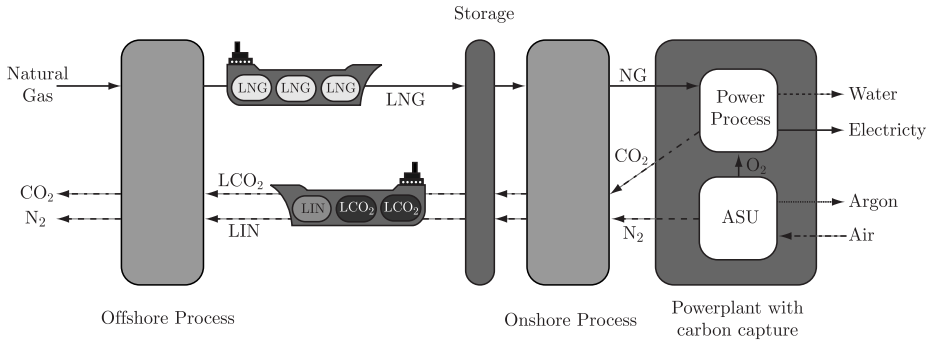


Figure B.8: The Liquefied Energy Chain

with power and utilities using the cold exergy from liquid carbon dioxide and liquid nitrogen, is optimized.

B.5.2 Design of an LNG process using LCO_2 and LIN as cold carriers

The Liquefied Energy Chain (LEC) [7] is a novel energy- and cost-effective transport chain for stranded natural gas that is utilized for onshore power production with CO_2 capture and offshore enhanced oil recovery (EOR). It includes an offshore section, a combined gas carrier, and an integrated receiving terminal, see Figure B.8. In the offshore section, natural gas is liquefied to produce LNG while Liquid Carbon Dioxide (LCO_2) and Liquid Inert Nitrogen (LIN) act as cold carriers. The reheated nitrogen is emitted to the atmosphere at ambient conditions while the CO_2 is transferred at high pressure to an offshore oilfield for EOR. LNG is transported to the receiving onshore terminal in the combined carrier. There, the cold exergy of the LNG is recovered in a liquefaction process for carbon dioxide and nitrogen. In this transport chain, CO_2 can be provided by industrial sources such as cement production, petrochemical plants or any power plant with CO_2 capture.

In a fully integrated energy chain, the onshore process is connected to an air separation unit that produces nitrogen for the offshore process and oxygen for an oxy-fuel power plant where natural gas is combusted to produce electricity as well as carbon dioxide and water. Water is removed from the flue gas. The CO_2 is compressed to a pressure above the triple point and liquefied by vaporization of the remaining LNG.

The LEC has better exergy efficiency and it is reasonable to believe that it will have lower investment costs than existing technology for dedicated transport of LNG and LCO_2. Furthermore, the concept shows potential for utilization of stranded natural gas with CO_2 sequestration on a commercially sound basis [7, 8].

In this example, it is shown how the ExPAnD methodology [9] and the previously discussed optimization formulation can be used to improve the design and optimize the operation of the offshore LNG process shown in Figure B.9. The goal is to design a process that is self-sufficient in the sense that it does not require the supply of utilities or work because space is at a premium in any offshore process.

Case	LNG [kg/s]	LIN [kg/s]	LCO_2 [kg/s]	W [kW]	Q_H [kW]	Q_C [kW]	Ψ [%]
I	1.0	1.83	2.46	820.2	888.6	0.0	57.9
II	1.0	1.29	2.46	−15.43	172.8	0.0	74.4
IIIa	1.0	0.0	2.46	114.05	15.99	467.76	52.1
IIIb	1.0	0.30	2.46	0.0	136.19	385.19	59.3
IIIc	1.0	0.90	2.46	0.0	24.359	0.0	84.9
IIId	1.0	0.90	2.46	0.0	0.0	0.0	84.6

Table B.6: Main results for LNG case study. I refers to the base case design, II after application of the ExPAnD methodology, IIIa–d refer to the different optimization scenarios. $W > 0$ indicates that work needs to be supplied while $W < 0$ means that work is generated.

Natural gas at 7 MPa and 15 °C is to be liquefied and let down to 0.1 MPa and -164.1 °C. The state of LCO_2 is to be changed from 0.55 MPa and -54.5 °C to 10 °C and 15 MPa, whereas LIN at 0.6 MPa and -177 °C is to be heated, vaporized and vented to the atmosphere at 0.1 MPa; the outlet temperature is not specified. The gas composition of the natural gas stream, ambient conditions and equipment data are as in [8]. The process design calculations are based on a production rate of 1 kg/s LNG. In the case of complete carbon capture, combustion of this natural gas stream will result in 2.73 kg/s CO_2. If one assumes that a practically feasible solution may capture 90% of the generated carbon dioxide, the flowrate of LCO_2 to the offshore LNG process is equal to 2.46 kg/s.

The process is simulated with HYSYS using the SRK equation of state. Figure B.10(a) shows the composite curves (CCs) for the process before any pressure manipulation is performed, which is referred to as Case I. Note that it is necessary to supply hot utility, which is provided by sea water available at ambient conditions. At 57.9%, the exergy efficiency of this process is low due to the large driving forces between the CCs and the energy intensive compression of CO_2 in the gas phase. The flowrate of LIN is minimized while meeting the requirement of no cold utility usage. Two reasons lead to this change in the considered objective. Firstly, the process is to operate on an offshore platform where space restrictions favor a process design that does not require utilities. These constraints, that are introduced in the studied cases below, lead to a meaningless objective if Equation (B.5) were still used. Secondly, the only task liquid nitrogen performs in this process is to provide exergy to the liquefaction train, but, at the same time, it takes up space in the cold carrier that could otherwise be used to ship CO_2. Additionally, a lot of power is required to produce liquid nitrogen in the onshore process. Thus, minimizing the nitrogen flowrate while meeting utility constraints leads to the most economical solution.

For completeness, the flowrates of the various streams, the required or produced work as well as the exergy conversion efficiency are shown in Table B.6. Note the sign convention for W: A positive value indicates that work needs to be supplied while a negative value means that work is generated.

The first step in the design procedure is to use the ExPAnD methodology [9] for streams that undergo a phase transition to develop an improved initial design, referred to as Case II.

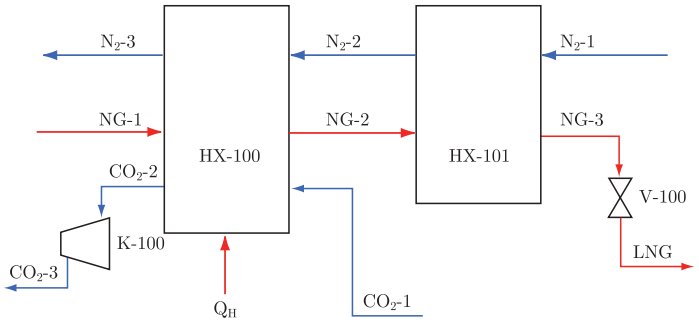
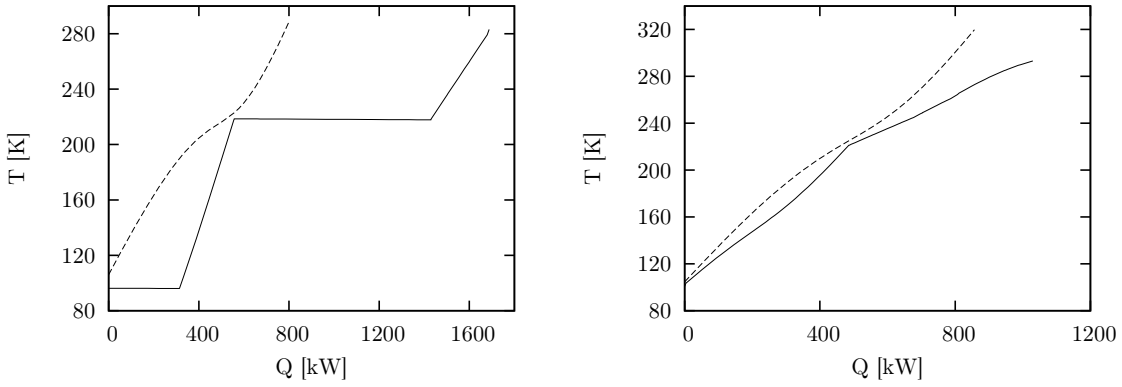


Figure B.9: Process flow diagram of the base case offshore LNG process before pressure manipulation



(a) CC before pressure manipulation (Case I)

(b) CC after application of ExPAnD (Case II)

Figure B.10: Composite curves for the offshore LNG process before pressure manipulation (a) and after application of the ExPAnD methodology (b)

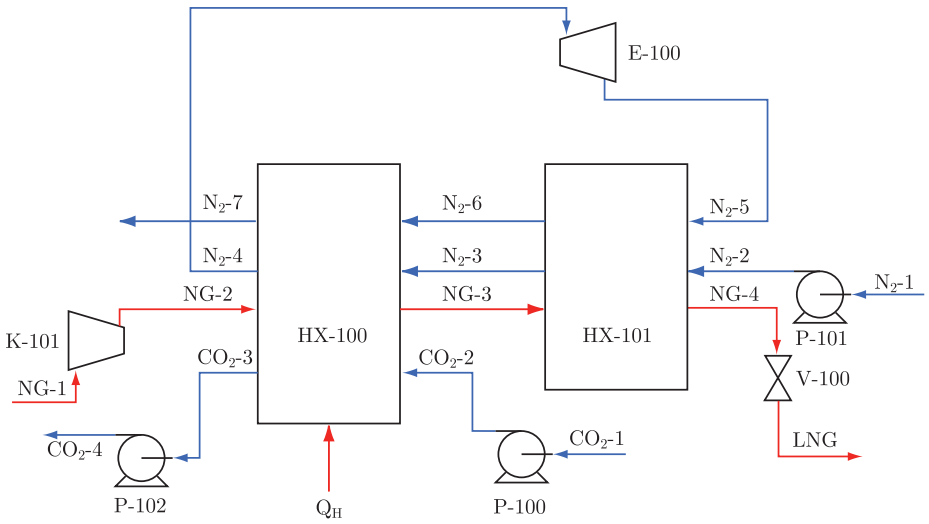


Figure B.11: Process flow diagram of the offshore LNG process after applying the ExPAnD methodology

The heuristics suggest that LCO_2 should be pumped to avoid compression. Likewise, the LIN should be pumped to 10 MPa to avoid the large driving forces in the heat exchanger, thereby transforming temperature-based exergy to pressure-based exergy. The nitrogen stream is then expanded and fed back to the heat exchanger to transform the pressure exergy into work and cold duty at a more appropriate temperature level. Finally, it can be shown that the natural gas should be compressed to 10 MPa before entering the heat exchangers to decrease the heat capacity flow rate of the natural gas stream in the pinch region. The new PFD and CCs are found in Figure B.11 and Figure B.10(b), respectively. Again, the process is simulated with HYSYS using the SRK equation of state. The required amount of LCO_2 and LIN, the net work produced and the exergy conversion efficiency are found in Table B.6. As the composite curves show, the varying heat capacity curve of the natural gas can be tracked much more efficiently leading to a steep increase in exergy efficiency to 74.4% while decreasing the nitrogen flowrate by 29.5%. For future reference, it should be noted that pumps P-100, P-101 and P-102 require 15.32 kW, 17.61 kW/(kg/s) F_{N_2} and 40.04 kW, respectively, and compressor K-101 uses 58.69 kW.

In the next step, the expansion of N_2 will be optimized to provide cold utility at the temperature levels necessary in order to reduce the required nitrogen flowrate again (Case III). Following the discussion earlier in this article, a cold stream with varying pressure levels is to be heated, expanded, heated, compressed, cooled, expanded and heated, as this will result in the best trade-off between increases in capital investment and process efficiency. The presented optimization formulation will be used to find the intermediate temperatures and pressures that will result in the smallest nitrogen flowrate.

The stream data for this initial design (inlet and outlet temperatures as well as averaged heat capacity values) are collected from the HYSYS model for Case II. Since the natural

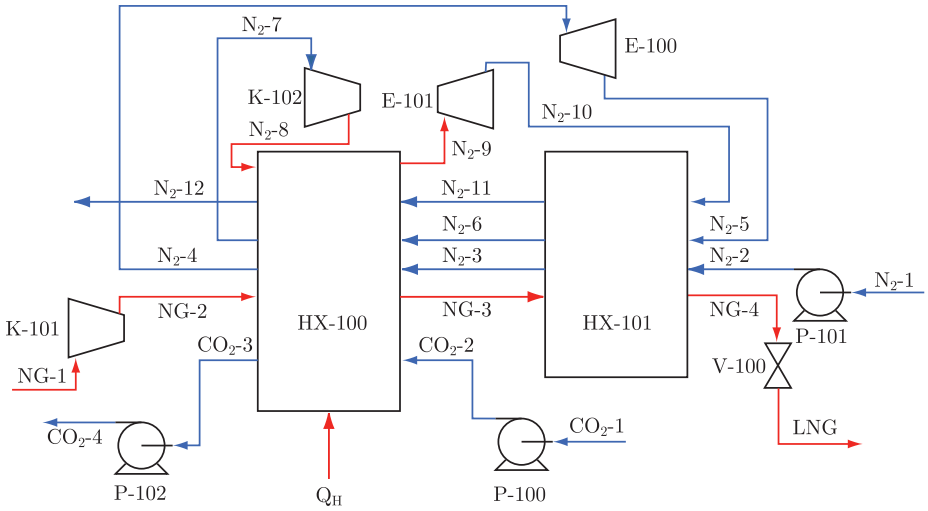


Figure B.12: Final process flow diagram for the offshore LNG process

gas consists of several components, mostly methane, ethane and propane, condensation will occur over a temperature interval. In the process, the natural gas stream is cooled at a pressure that is above the critical point. The temperature/enthalpy curve will therefore not have the flat condensation region normally found when condensing a single component fluid. However, the heat capacity is still far from constant. Therefore, the stream is divided into three individual streams (H1–H3), which yield a reasonably good fit to the actual cooling curve. Similarly, the heat capacity flowrate of the high-pressure liquid nitrogen and liquid CO_2 streams are not constant either. The liquid carbon dioxide stream is divided into two individual streams (C1–C2). Its inlet and outlet temperatures as well as the pressure and flowrates remain fixed. The high-pressure nitrogen stream to be expanded is treated as a variable stream according to Figure B.3 allowing for two expansion cycles and one compression cycle with intermediate heating and cooling, resulting in three possible cold streams and one possible hot stream. The initial cold nitrogen stream is split into three streams (C3–C5) to get more accurate fit of the averaged heat capacities. Expansion, compression and expansion result in streams C6, H4 and C7, respectively. The outlet temperature of the nitrogen stream is variable. Overall, the process is modeled using a total of four hot and seven cold streams with three possible pressure manipulations; see Figure B.12.

Due to the high pressure and low temperature of the nitrogen stream, the first expansion is far from ideal; hence a non-ideal polytropic exponent of $\kappa = 1.51$ together with an efficiency factor for the work of $\eta_C = 0.7$ are used for the first expansion, based on a comparison with the HYSYS simulation. For the possible expansions and compressions below 4 MPa the pressure operator, which is based on the ideal gas model, is accurate. The hot and cold utility temperatures are set to 383.15 K and 93.15 K, respectively. Here, $\Delta T = 4\,\mathrm{K}$, $\kappa = 1.352$, $\eta_C = \eta_E = 1.0$, $M = 400\,\mathrm{K}$, $U = 1300\,\mathrm{kW}$ and $\varepsilon = 0.1\,\mathrm{K}$. The pressure of C6 is constrained to be between 0.3 and 1 MPa, while the one of H4 can vary between

Stream	F_s [kg/s]	$c_{p,s}$ [kJ/kg]	$T_{s,\text{in}}$ [K]	$T_{s,\text{out}}$ [K]	p_s [MPa]
H1 (NG-2–NG-4)	1.0	3.46	319.8	265.15	10.0
H2 (NG-2–NG-4)	1.0	5.14	265.15	197.35	10.0
H3 (NG-2–NG-4)	1.0	3.51	197.35	104.75	10.0
H4 (N_2-8–N_2-9)	—	1.15	—	—	—
C1 (CO_2-2–CO_2-3)	2.46	2.11	221.12	252.55	6.0
C2 (CO_2-2–CO_2-3)	2.46	2.48	252.55	293.15	6.0
C3 (N_2-2–N_2-4)	—	2.48	103.45	171.05	10.0
C4 (N_2-2–N_2-4)	—	1.80	171.05	218.75	10.0
C5 (N_2-2–N_2-4)	—	1.18	218.75	—	10.0
C6 (N_2-5–N_2-7)	—	1.07	—	—	—
C7 (N_2-10–N_2-12)	—	1.04	—	—	0.1

Table B.7: Given data for the optimization of the offshore LNG process

1.0 and 3.5 MPa. The flowrates of the nitrogen streams are equal throughout the flowsheet; similarly, streams consisting of carbon dioxide have equal flowrate. Table B.7 shows the stream data for the optimization model. There are seven decision variables: the nitrogen flowrate (F_{N_2}), the intermediate outlet temperatures ($T_{H4,out}$, $T_{C5,out}$, $T_{C6,out}$) and pressures (p_{H4}, p_{C6}) of the nitrogen streams as well as the outlet temperature of the nitrogen stream ($T_{C7,out}$). The goal is to design a process that is self-sustained, i.e., that does not require utilities or work, with the minimal nitrogen flowrate. Thus, the objective function is to minimize the flowrate of nitrogen.

The following cases are investigated:

- Case IIIa: minimize flowrate of nitrogen,
- Case IIIb: minimize flowrate of nitrogen so that $W \leq 0$,
- Case IIIc: minimize flowrate of nitrogen so that $Q_C = 0$ and $W \leq 0$,
- Case IIId: minimize flowrate of nitrogen so that $Q_C = Q_H = 0$ and $W \leq 0$.

In Case IIIa, the minimal N_2 flowrate is found using one hot and one cold utility available at 383.15 K and 93.15 K, respectively, in accordance with the previously described optimization model. As can be expected in the presence of utilities and external sources of work, no nitrogen is required to support the process. As can be seen from the composite curves, which are not balanced with utilities, in Figure B.13(a), CO_2 provides only a very narrow temperature interval heat sink for the natural gas stream. Since there is no N_2 stream, no work is produced by expanding it and the process requires that 114.05 kW of work is supplied. Overall, these factors lead to a low exergy efficiency of 52.1%; see Table B.6. The problem was solved in 5 seconds. It should be pointed out that the optimization formulation in this case does not find the given utilities. Instead, due to degeneracy with respect to the objective of minimizing the nitrogen flowrate, it reports

Variable	Unit	Case IIIa	Case IIIb	Case IIIc	Case IIId
F_{N_2}	[kg/s]	0.0	0.296	0.898	0.898
$T_{H4,in}$	[K]	319.9	383.15	365.01	365.00
$T_{H4,out}$	[K]	319.8	383.15	225.12	226.56
p_{H4}	[MPa]	3.50	3.50	2.677	2.743
$T_{C5,out}$	[K]	218.75	383.15	222.55	221.11
$T_{C6,in}$	[K]	97.89	210.39	95.66	95.66
$T_{C6,out}$	[K]	188.23	383.15	221.12	221.15
p_{C6}	[MPa]	0.457	1.00	0.390	0.400
$T_{C7,in}$	[K]	123.73	210.39	95.66	95.66
$T_{C7,out}$	[K]	315.7	261.05	383.15	357.05

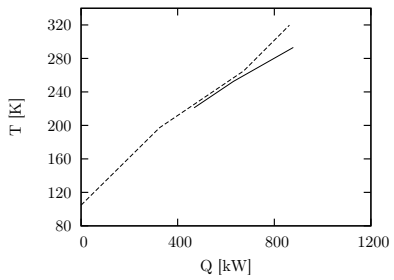
Table B.8: Results for the decision variables for Case III of the LNG offshore process design

results that are increased by an arbitrary constant. However, the minimum utility needed can be calculated easily. The values of the decision variables for all Cases IIIa–IIId are shown in Table B.8.

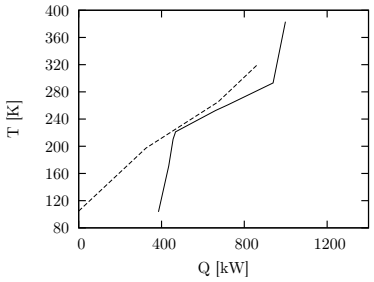
In Case IIIb, an additional constraint is added requiring that the process does not require work, i.e., $W \leq 0$. As a result, the flowrate of nitrogen at the optimal solution is increased though still only a fraction of what had been found in cases I and II. The required cold utility is decreased in comparison to the previous result, see Table B.6. In this case, at the found solution, both p_{C6} and p_{H4} are at their upper bounds, see Table B.8. In order to provide the required work, the inlet temperature of the second expander on the nitrogen stream is chosen as large as possible. This is achieved by heating C5 with utility and by not cooling stream H4. Overall, the exergy efficiency of the design increases to 59.3%, which is still below the results found in Cases I and II. The solution was found in 143 seconds.

Case IIIc solves the problem with an additional constraint forcing $Q_C = 0$. To make up for the lack of cold utility, the flowrate of N_2 supplied to the process is increased over the previous two cases. Only a small amount of hot utility is required which is basically used only to heat the vented nitrogen stream to its high outlet temperature. As can be seen from Figure B.13, the cold composite curve is able to track the cooling curve of the natural gas nicely. Additionally, the exergy efficiency is increased to 84.9% and surpassed the values found in the early designs (Cases I and II). The solution was found in 13 hours and 8 minutes.

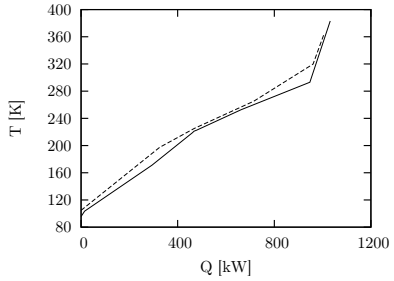
Case IIId adds the constraint that no hot utility may be used, i.e., $Q_H = 0$. The resulting process design is very similar to the results from the previous case, as can be seen in Table B.8. This can be explained as follows: The objective function is not impacted when $T_{C7,out}$ is varied a within certain range, but the necessary hot utility Q_H changes, thus creating a degenerate optimal solution. Fixing $Q_H = 0$ removes this degeneracy from the problem and reduces the computational effort, too. The solution was found in 3 hours and 26 minutes. Note that the exergy efficiency is slightly smaller than in the previous result. This results from the reduced exhaust temperature of the vented N_2 stream, which



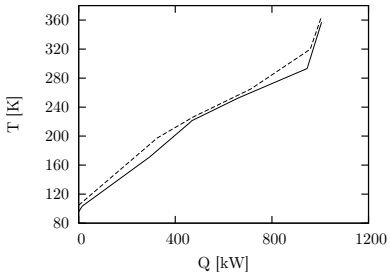
(a) Case IIIa: minimum N_2 flowrate



(b) Case IIIb: minimum N_2 flowrate with $W \leq 0$



(c) Case IIIc: minimum N_2 flowrate with $W \leq 0$ and $Q_C = 0$



(d) Case IIId: minimum N_2 flowrate with $W \leq 0$ and $Q_C = Q_H = 0$

Figure B.13: Compositive curve for the offshore LNG process resulting from the different optimization cases

exergy content can be considered lost anyway and is only included for completeness. In comparison to Case II, the nitrogen flowrate could be reduced by 30.4%; in comparison to Case I, it is reduced by 50.8%. Furthermore, the process does not require any supplied utility or work, see Table B.6.

Due to the simplifications used in the model formulation, the results cannot directly be implemented in HYSYS when a more rigorous physical property model is selected in the process simulator. For example, given the same pressure difference across an expander and the same temperature at the expander inlet, a reversible process with an ideal gas will result in a different outlet temperature than a reversible process modeled with a cubic equation of state. Similarly, the hot and cold streams are modeled to have constant heat capacities, while HYSYS models the streams more accurately where heat capacity is a function of temperature. Thus, it is necessary to change the intermediate temperatures and pressures found by the model slightly to obtain a feasible result upon implementation of the optimization results in HYSYS.

B.6 Discussion

It is shown that by expanding and compressing process streams appropriately (according to the plus–minus principle in PA) the requirements for hot and cold utilities may be significantly reduced. A superstructure for such pressure manipulations is developed, showing that a hot stream may change to a cold stream after expansion and that a cold stream may shift to a hot stream upon compression. If possible, a process stream should be compressed or expanded from the pinch temperature given that the pinch point does not change. Since it is likely that the pinch point will change, however, optimization is required.

Allowing for compression and expansion of the process streams will increase the complexity of traditional PA significantly as one stream will result in several streams with the possibility for expansions and compressions. Furthermore, as the inlet temperatures to the pinch operator will vary, it is necessary to use the pinch operator suggested by Grossmann et al. [71]. This pinch operator is nonlinear in the case of varying heat capacity flowrates. To model the second law constraints in the pinch operator, it requires a large number of binary variables leading to a nonconvex MINLP. In addition, a pressure operator and an exergy operator are developed, both are based on ideal gas assumptions. The pressure operator introduces additional nonlinear and nonconvex terms, hence combining the operators (pinch, pressure, exergy) forms a nonconvex MINLP, which needs to be solved by a global solver. By combining the operators, the minimal irreversibilities for a heat exchanger network that allows for compression and expansion can be solved. Due to the nonconvexity of the MINLP formulation, however, only small problems can be solved at present.

In the examples, a maximum of three pressure manipulations are allowed; however, by adding more heat exchanger passes, compressors and expanders, even higher thermodynamic efficiencies can be obtained. The marginal effect of adding additional expanders

and compressors is not expected to justify the additional capital investments. If the process inlet and outlet specifications are constant, it is possible to find the minimal irreversibilities by minimizing the net exergy input from the hot and cold utilities and maximizing the work produced, as done in the first example. If the heat capacity flowrates of the process streams are allowed to vary, the exergy conversion efficiency, including the inlet and outlet exergy of the process streams must be maximized to obtain good results.

From the first studied example it can be seen that it is possible to obtain a solution (Case 4) where both hot and cold utilities are avoided; such a solution cannot be expected for every problem, though. If $\Delta T_{\min}$ in Case 2 had been increased it would not have been possible to find a solution without hot or cold utilities and no net work. The exergy efficiency would decrease due to the increased temperature differences (and thereby larger irreversibilities), however, the costs (area) of the heat exchangers would also decrease. For ideal expansions and compressions it is difficult to obtain a solution without utilities and with no net work without changing the flowrate of the process streams. This is easier to achieve with compressor and expander isentropic efficiencies less than 100%, as there will be losses in each compressor/expansion cycle which are dependent on the inlet temperature.

As can be seen from Table B.6, the proposed process of using liquid nitrogen and liquid CO_2 to liquefy natural gas will have a very low thermodynamic efficiency if the pressures of the process streams are not manipulated. The most important contribution to increase the efficiency comes from using sound engineering knowledge formalized in the ExPAnD methodology. However, the selection of the intermediate pressure and temperature levels in the nitrogen loop is not so straightforward. It is here where the optimization formulation delivers additional value. It suggests a process design that satisfies the constraints of no utility usage and further lowers the N_2 flowrate.

Producing hot and cold utilities as well as work can be very expensive in an offshore process; hence it is shown how these utilities can be avoided by compression and expansion of process streams. Also, since the net work produced cannot necessarily be used at a field site location, it should not be accounted for as useful exergy. Therefore, in Case IIId, constraints are added so that the process is to be self-sustained, that is, without hot or cold utilities and without producing or consuming power. A more thorough description of the processes in the liquefied energy chain can be found in [8].

There are three main challenges with the proposed model. Firstly, since the problem is a nonconvex MINLP, a global solver, such as BARON or nonconvex outer approximation [97] must be used to find the global optimum. This has the implication that only small problems such as those considered here can be solved within a reasonable time at present, even with reasonable upper and lower bounds for the variables. Secondly, it is not easy to set appropriate bounds without having in-depth knowledge of the process to be optimized; hence the design tends to be an iterative process. Finally, due to the simplifications required for the model, e.g., constant heat capacity flowrate and ideal gas assumptions, the solution found by the global solver may prove not to be feasible in HYSYS simulations, since HYSYS has access to more rigorous thermodynamic models, and thereby gives a more accurate estimate for the process to be designed in the real world. On the other hand,

minor adjustments can be made to eliminate this infeasibility.

B.7 Conclusions

An optimization formulation for heat and power integration is developed and implemented in GAMS using BARON as the global solver. In this extended problem definition, the process streams are allowed to undergo pressure changes as well as phase and temperature changes. The procedure is particularly suited for subambient processes where pressure-based exergy can be transformed into temperature-based exergy and vice versa by expansion and compression. The resulting design consists of heat exchangers, pumps, compressors and expanders integrated in a way that minimizes total irreversibilities. To design less complex and less costly processes, constraints can be added that disallow (if at all possible) the use of external heating and cooling as well as external power. The proposed approach combines Pinch Analysis, Exergy Analysis and Mathematical Programming (a nonconvex MINLP model). It should be stressed that the problem addressed and solved by this new Process Synthesis tool is significantly more complex than the traditional Heat Exchanger Network Synthesis problem. The examples show that manipulation of stream pressures can significantly reduce the total irreversibilities in Heat Exchanger Networks. An industrial application related to LNG shows that the optimization formulation is capable of suggesting a reasonable initial design for realistic problems.

Although the proposed optimization model can give a reasonable design for new processes, the formulation can be improved and expanded to be even more general. First of all a more sophisticated pressure operator based on more accurate equations of state, for example SRK, should be implemented to achieve more accurate results for non-ideal gases. Also, equations for liquid pumping and liquid expansion should be included. Finally, equations for phase transitions should be added. However, since the model already has reached the limit for how large problems one can solve, this is not considered at the current stage.

Appendix C

Pinch operator for streams with non-constant heat capacity

Two problems in heat exchanger network synthesis are considered, the utility targeting problem and heat exchanger network synthesis problem. In contrast to most literature methods, constant heat capacity is not assumed here.

Let S denote the set of *process streams*. A process stream $i \in S$ is a stream with flowrate F_i, heat capacity $c_{p,i}(T)$, and in- and outlet temperature $T_{\text{in},i}$ and $T_{\text{out},i}$, respectively. A process stream i is a *hot process stream* when $T_{\text{in},i} \geq T_{\text{out},i}$; similarly, it is termed a *cold process stream* if $T_{\text{in},i} \leq T_{\text{out},i}$. Let n_H and n_C be the number of hot and cold process streams and denote their respective sets by H and C; $H \cap C = S$. Furthermore, assume that a hot and a cold utility is available.

The *utility targeting problem* [e.g., 83, 108, 109, 132] can be described as follows: Given a set of hot and cold streams, H and C, determine the minimum hot and cold utility heat loads, Q_H and Q_C, respectively, so that first and second law constraints are satisfied. Additionally, a minimum temperature difference ΔT_{min} between hot and cold streams is enforced. In the case of constant heat capacity considered in the referenced papers, this problem leads to a linear program.

The *heat exchanger network synthesis* problem [44, 62, 77, 134, 179–181] is more complex. Given a set of hot and cold streams, H and C, determine the optimal network of heat exchangers that leads to minimal cost in some cost measure, which typically includes both investment and operating expenses. Here, in addition to finding the necessary utility duties, it is also necessary to identify matches between hot and cold streams, to determine if process streams need to be split, and to size the equipment. This problem leads to a mixed-integer nonlinear program.

C.1 Utility targeting for streams with non-constant heat capacity

In this section, a novel method to obtain utility targets is described. In contrast to previous methods, it is able to provide targets for problems where the heat capacity of streams are non-constant. In literature formulations, it is assumed that the heat capacity is constant [e.g., 83, 108, 109, 132].

Pinch analysis popularized by [108, 109] aggregates the information of the individual streams into composite curves for the hot and cold streams, $T_H(Q)$ and $T_C(Q)$; cf. Fig. C.1.

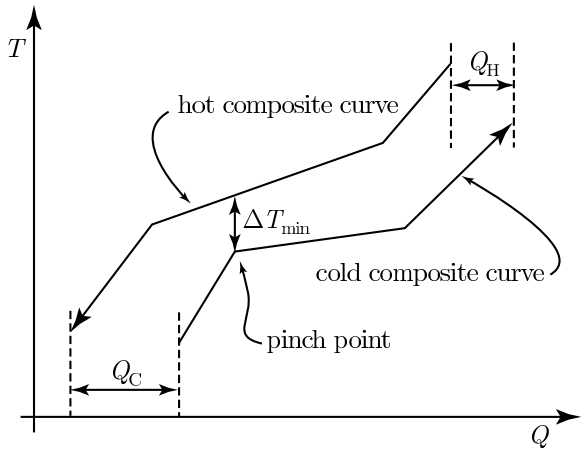


Figure C.1: Illustration of the basic concept of pinch analysis in the case of constant heat capacity flowrates

Then, one shifts one of the curves horizontally so that each point on the cold composite curve is always at least $\Delta T_{\min}$ below the hot composite curve in order to guarantee second law feasibility. The point, at which the distance $T_H(Q) - T_C(Q) \geq 0$ is minimal, is termed *pinch point*. The key insight that enabled efficient methods was discovered by Hohmann [83]. He pointed out that the pinch point can only occur at the inlet temperature of process streams *if the heat capacity of the streams are constant*. Thus, optimization formulation for this problems need only guarantee second law feasibility at a finite number of points.

Throughout the literature it is argued that nonconstant heat capacities can be approximated by splitting the stream into parts with constant heat capacities [55]. However, Castier and Queiroz [39] remark that nonlinear behavior is not just a pathological case and Hasan et al. [76] point out that streams undergoing phase changes or close to the critical point also lead to varying heat capacities. Here, a optimization formulation is proposed that allows for process streams with varying heat capacities.

Define the respective *aggregated* hot and cold stream *inlet* and *outlet temperatures* as $T_{\mathrm{in},H} = \max\{T_{\mathrm{in},i} | i \in H\}$, $T_{\mathrm{out},H} = \min\{T_{\mathrm{out},i} | i \in H\}$, $T_{\mathrm{in},C} = \min\{T_{\mathrm{in},i} | i \in C\}$, and $T_{\mathrm{out},C} = \max\{T_{\mathrm{out},i} | i \in C\}$. Furthermore, define the *aggregated* hot and cold stream *flowrate* and *heat capacities* as

$$F_H(T) = \sum_{i \in H | T \in [T_{\mathrm{out},i}, T_{\mathrm{in},i}]} F_i, \qquad c_{p,H}(T) = \sum_{i \in H | T \in [T_{\mathrm{out},i}, T_{\mathrm{in},i}]} c_{p,i}(T),$$
$$F_C(T) = \sum_{i \in C | T \in [T_{\mathrm{in},i}, T_{\mathrm{out},i}]} F_i, \quad \text{and} \qquad c_{p,C}(T) = \sum_{i \in C | T \in [T_{\mathrm{in},i}, T_{\mathrm{out},i}]} c_{p,i}(T).$$

The notation states that, for each T, the summation includes process streams which are present at T.

The first law constraint for the heat exchanger network is straightforward to obtain. An

overall energy balances yields that

$$Q_C + \int_{T_{\text{in},C}}^{T_{\text{out},C}} F_C(T)c_{p,C}(T)\,dT = Q_H + \int_{T_{\text{out},H}}^{T_{\text{in},H}} F_H(T)c_{p,H}(T)\,dT. \tag{C.1}$$

Clearly, it determines a linear relationship between Q_H and Q_C. Similarly, a second law constraint can be formulated. It is necessary to guarantee that the heat exchange is feasible, i.e., heat is always transfered from a higher to lower temperature and from hot to cold stream. Appealing to the often used depiction from pinch analysis, see Fig. C.1, the hot stream always needs to be above the cold stream. Since typically available data is heat capacity as a function of temperature, consider a figure where abscissa and ordinate are reversed, see Fig. C.2. This figure also illustrates the formulation for the second law constraints. First, note that $F_H(T)c_{p,H}(T)$ can be interpreted as the slope of the hot composite curve in this illustration. Since only energy differences are important in this context, suppose that $(T_{\text{out},H}, 0)$ is a point on the hot composite curve. Let $(T_{\text{in},C}, Q_C)$ be a point on the cold composite curve. Feasibility implies that the cold composite curve is always *above* the hot composite curve. Thus, one needs to find the minimal Q_C, i.e., the smallest vertical shift of the cold composite curve, so that the feasibility requirement is met. This requirement can formally be written as

$$\Delta Q(T^*) \equiv Q_C + \int_{T_{\text{in},C}}^{T^*} F_C(T)c_{p,C}(T)\,dT - \int_{T_{\text{out},H}}^{T^*+\Delta T_{\min}} F_H(T)c_{p,H}(T)\,dT \geq 0, \quad \forall T^* \in \Theta, \tag{C.2}$$

where $\Theta = [\max\{T_{\text{in},C}, T_{\text{out},H} - \Delta T_{\min}\}, \min\{T_{\text{out},C}, T_{\text{in},H} - \Delta T_{\min}\}]$. Thus, the targeting problem can be formulated as

$$\begin{aligned} \min_{Q_H, Q_C} \quad & c_H Q_H + c_C Q_C \\ \text{s.t.} \quad & \text{(C.1)}, \\ & \text{(C.2)}, \\ & Q_H, Q_C \geq 0, \end{aligned} \tag{TP}$$

where c_H and c_C are the specific costs of cold and hot utility. This is a semi-infinite program (SIP) due to constraint (C.2) which needs to hold for each T in an interval.

Obviously, the definition of aggregated flowrates and heat capacities leads to piecewise linear and piecewise smooth functions, respectively. It is advantageous to consider each temperature interval, on which F_H and F_C are constant and $c_{p,H}$ and $c_{p,C}$ are smooth, individually. Therefore, divide Θ into subintervals at each $T_{\text{in},i}, T_{\text{out},i} \in \Theta$ where $i \in C$ and $T_{\text{in},i} + \Delta T_{\min}, T_{\text{out},i} + \Delta T_{\min} \in \Theta$ where $j \in H$. This leads to at most $2n_H + 2n_C - 2$ subintervals Θ_j, $j = 1 \ldots, l$. Let $\theta_j = \inf\{T | T \in \Theta_j\}$, $j = 1, \ldots, l$ and define $q_{H,1} =$

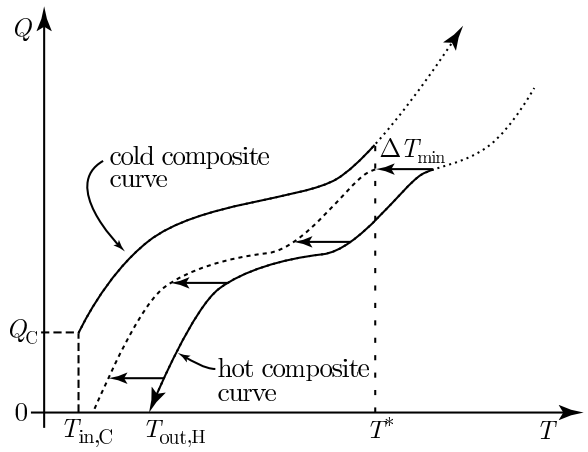


Figure C.2: Illustration of the second law constraint for feasibility of the heat exchange for process streams with nonconstant heat capacity. If for all T^* the cold composite curve is above the dotted curve, which is the hot composite curve shifted left by $\Delta T_{\min}$, then the heat exchange is feasible for a given Q_C. Note that ordinate and abscissa are swapped here in contrast to the traditional construction in Figure C.1.

$\int_{T_{\text{out},H}}^{\theta_1} F_H(T)c_{p,H}(T)\,dT$ and $q_{C,1} = \int_{T_{\text{in,C}}}^{\theta_1} F_C(T)c_{p,C}(T)\,dT$. Let

$$q_{H,j+1} = q_{H,j} + \int_{\theta_j}^{\theta_{j+1}} F_H(T)c_{p,H}(T)\,dT,\ j = 1,\dots,l-1 \quad \text{and}$$
$$q_{C,j+1} = q_{C,j} + \int_{\theta_j}^{\theta_{j+1}} F_C(T)c_{p,C}(T)\,dT,\ j = 1,\dots,l-1.$$

Note that these parameters can be computed in advance.

Thus, the semi-infinite constraint for $\Delta Q(T)$, (C.2), can be reformulated as l semi-infinite constraints

$$\Delta Q_j(T^*) \equiv Q_C + q_{C,j} + \int_{\theta_j}^{T^*} F_C(T)c_{p,C}(T)\,dT - q_{H,j} - \int_{\theta_j}^{T^*+\Delta T_{\min}} F_H(T)c_{p,H}(T)\,dT \geq 0, \tag{C.3}$$

which need to hold for all $T^* \in \Theta_j$, $j = 1,\dots,l$. In contrast to ΔQ, the integrands are continuous functions now and their functional description does not change, e.g., one has $F_H(T) = F_{H,j}$ and $F_C(T) = F_{C,j}$ for each $T \in \Theta_j$.

C.1.1 Reformulating the targeting problem

In the previous section, it was shown that the targeting problem can be formulated as a SIP when the assumption of constant heat capacity is dropped. Bhattacharjee et al. [32] describe a method to find a global minimum of a SIP.

Finding global optimal solutions of SIPs using the method in [32] requires the construction of a lower-bounding and an upper-bounding problem. A valid lower-bound can be found by replacing the interval Θ_j with a finite set, $\Xi_j \subset \Theta_j$. The lower bounding problem can be written as

$$\begin{aligned} \min_{Q_H, Q_C} \quad & c_H Q_H + c_C Q_C \\ \text{s.t.} \quad & \text{(C.1)}, \\ & \Delta Q_j(T^*) \geq 0, \forall T^* \in \Xi_j,\ j = 1, \ldots, l, \\ & Q_H, Q_C \geq 0. \end{aligned} \tag{LBP}$$

Since only a finite number of constraints of (TP) are present in (LBP), the feasible space of the latter problem is a superset of the feasible space of the former problem, which indicates that it is indeed a lower-bounding problem.

Constructing an upper-bounding problem is more involved, see the discussion in [32]. It requires finding an inclusion bound $\Delta Q_j^L(\Theta_j) \leq \min_{T \in \Theta_j} \Delta Q_j(T)$ where $\Delta Q_j^L(\Theta_j)$ depends on the lower and upper bound of Θ_j. The upper bounding problem can be written as

$$\begin{aligned} \min_{Q_H, Q_C} \quad & c_H Q_H + c_C Q_C \\ \text{s.t.} \quad & \text{(C.1)}, \\ & \Delta Q_j^L \geq 0,\ j = 1, \ldots, l, \\ & Q_H, Q_C \geq 0. \end{aligned} \tag{UBP}$$

For the purpose of the following discussion, it is assumed that $c_p(T) = a + bT + cT^2 + dT^3$, which is also the functional form[1] for which ideal gas heat capacity data is collected in [141]. Let $a_{H,j}$ and $a_{C,j}$ denote the first parameter in the aggregated heat capacity of the hot and cold stream, respectively. Similar symbols are introduced for the other parameters. Given a cubic expression for c_p, the analytical solution of the integrals in (C.3) is

$$\begin{aligned} & F_{C,j} \int_{\theta_j}^{T^*} c_{p,C}(T)\, dT - F_{H,j} \int_{\theta_j}^{T^* + \Delta T_{\min}} c_{p,H}(T)\, dT \\ & \quad = F_{C,j} \left[\frac{1}{12} T (12 a_{C,j} + T(6 b_{C,j} + T(4 c_{C,j} + 3 d_{C,j} T))) \right]_{\theta_j}^{T^*} \\ & \qquad - F_{H,j} \left[\frac{1}{12} T (12 a_{H,j} + T(6 b_{H,j} + T(4 c_{H,j} + 3 d_{H,j} T))) \right]_{\theta_j}^{T^* + \Delta T_{\min}}. \end{aligned} \tag{C.4}$$

A valid lower bound for this expression needs to be constructed. Inclusion bounds for the factorable function that result from the integration as demonstrated in (C.4) can be

[1] It should be noted however that the method discussed here is not restricted to this choice. On the contrary, it is only required that it is possible to find the analytical solution of the integrals in (C.3).

constructed using these expressions

$$\underline{v}_{j,1} = 4c + 3\min\{d\underline{T}, d\overline{T}\}, \qquad \overline{v}_{j,1} = 4c + 3\max\{d\underline{T}, d\overline{T}\},$$
$$\underline{v}_{j,2} = 6b + \min\{\underline{T}\,\underline{v}_{j,1}, \underline{T}\overline{v}_{j,1}, \overline{T}\underline{v}_{j,1}, \overline{T}\overline{v}_{j,1}\}, \qquad \overline{v}_{j,2} = 6b + \max\{\underline{T}\,\underline{v}_{j,1}, \underline{T}\overline{v}_{j,1}, \overline{T}\underline{v}_{j,1}, \overline{T}\overline{v}_{j,1}\},$$
$$\underline{v}_{j,3} = 12a + \min\{\underline{T}\,\underline{v}_{j,2}, \underline{T}\overline{v}_{j,2}, \overline{T}\underline{v}_{j,2}, \overline{T}\overline{v}_{j,2}\}, \qquad \overline{v}_{j,3} = 12a + \max\{\underline{T}\,\underline{v}_{j,2}, \underline{T}\overline{v}_{j,2}, \overline{T}\underline{v}_{j,2}, \overline{T}\overline{v}_{j,2}\},$$
$$\underline{v}_{j,4} = \frac{1}{12}\min\{\underline{T}\,\underline{v}_{j,3}, \underline{T}\overline{v}_{j,3}, \overline{T}\underline{v}_{j,3}, \overline{T}\overline{v}_{j,3}\}, \qquad \overline{v}_{j,4} = \frac{1}{12}\max\{\underline{T}\,\underline{v}_{j,3}, \underline{T}\overline{v}_{j,3}, \overline{T}\underline{v}_{j,3}, \overline{T}\overline{v}_{j,3}\}.$$

Note that the indices for the parameters for c_p are dropped to ease notation. It should be clear from the context which indices are meant.

Then, for all $T^* \in [\theta_j, \theta_{j+1}]$,

$$F_{C,j}\int_{\theta_j}^{T^*} c_{p,C}(T)\,dT \geq F_{C,j}\left[\underline{v}_{j,4} - \frac{1}{12}\theta_j(12a_{C,j} + \theta_j(6b_{C,j} + \theta_j(4c_{C,j} + 3d_{C,j}\theta_j)))\right] \tag{C.5}$$

where $\underline{T} = \theta_j$ and $\overline{T} - \theta_{j+1}$ and the parameters for the cold aggregate heat capacity are used. Similarly, for all $T^* \in [\theta_j + \Delta T_{\min}, \theta_{j+1} + \Delta T_{\min}]$,

$$F_{H,j}\int_{\theta_j}^{T^*} c_{p,H}(T)\,dT \leq F_{C,j}\left[\overline{v}_{j,4} - \frac{1}{12}\theta_j(12a_{H,j} + \theta_j(6b_{H,j} + \theta_j(4c_{H,j} + 3d_{H,j}\theta_j)))\right] \tag{C.6}$$

where $\underline{T} = \theta_j + \Delta T_{\min}$ and $\overline{T} - \theta_{j+1} + \Delta T_{\min}$ and the parameters for the hot aggregate heat capacity are used. Using the inequalities (C.5) and (C.6) in (C.2), a valid lower bound ΔQ_j^L for the feasibility constraint is obtained.

C.2 Heat exchanger network synthesis for streams with non-constant heat capacity

While in the previous section only the utility targeting problem was considered, here a formulation for the synthesis of heat exchanger networks for streams with non-constant heat capacity will be presented. It uses the superstructure introduced by [179] that considers stages at which, in principle, each hot stream can contact each cold stream.

In addition to the variables introduced in the beginning of this Chapter, some additional definitions are necessary. Let K be the set of stages of heat exchangers. At stage $k \in K$, hot stream i can possibly be contacted with cold stream j to transfer heat q_{ijk}. If this contact occurs, it is indicated by the binary variable $z_{ijk} = 1$, otherwise $z_{ijk} = 0$. Also, heat exchange with the utilities are considered: qcu_i is the heat transfered from the hot stream i to the cold utility, $zcu_i = 1$ indicates if this occurs; qhu_j is the heat transfered to the cold stream j from the hot utility, $zhu_j = 1$ indicates if this heat transfer occurs. The temperature of process stream $i \in S$ between stage k and $k+1$ is $t_{i,k+1}$. Lastly, Ω and Γ denote upper bounds on the transferred energy and temperature differences, respectively.

In the model isothermal mixing is assumed, since flow rates through each heat exchanger

in the stages need to be tracked otherwise which leads to a more complicated problem formulation.

An overall energy balance for each of the process streams yields

$$\int_{T_{\text{out},i}}^{T_{\text{in},i}} F_i c_{p,i}(T)\, dT = \sum_{k\in K}\sum_{j\in C} q_{ijk} + qcu_i, \quad \forall i \in H,$$
$$\int_{T_{\text{in},j}}^{T_{\text{out},j}} F_j c_{p,j}(T)\, dT = \sum_{k\in K}\sum_{i\in H} q_{ijk} + qhu_j, \quad \forall j \in C.$$

An energy balance of a process stream at each stage results in

$$\int_{t_{i,k+1}}^{t_{i,k}} F_i c_{p,i}(T)\, dT = \sum_{j\in C} q_{ijk}, \quad \forall i \in H, k \in K,$$
$$\int_{t_{j,k+1}}^{t_{j,k}} F_j c_{p,j}(T)\, dT = \sum_{i\in H} q_{ijk}, \quad \forall j \in C, k \in K.$$

The consumed utilities are calculated from

$$\int_{T_{\text{out},i}}^{t_{i,|K|+1}} F_i c_{p,i}(T)\, dT = qcu_i, \quad \forall i \in H$$
$$\int_{t_{j,1}}^{T_{\text{out},j}} F_j c_{p,j}(T)\, dT = qhu_j, \quad \forall j \in C.$$

The inlet temperatures of the process streams are assigned to the stage temperatures

$$t_{\text{in},i} = t_{i,1}, \quad i \in H,$$
$$t_{\text{in},j} = t_{j,|K|+1}, \quad j \in C,$$

and the outlet temperatures provide bounds on the stage temperatures

$$T_{\text{out},i} \geq t_{i,|K|+1}, \quad \forall i \in H,$$
$$T_{\text{out},j} \leq t_{j,1}, \quad \forall j \in C.$$

The stage temperatures decrease with increasing stage number

$$t_{i,k} \geq t_{i,k+1}, \quad \forall i \in H, k \in K,$$
$$t_{j,k} \geq t_{j,k+1}, \quad \forall j \in C, k \in K.$$

The following logical constraints are used to set the binary variables.

$$\begin{aligned} 0 \leq q_{ijk} &\leq \Omega\, z_{ijk}, \quad i \in H, j \in C, k \in K, \\ 0 \leq qcu_i &\leq \Omega\, zcu_i, \quad i \in H \\ 0 \leq qhu_j &\leq \Omega\, zhu_j, \quad j \in C. \end{aligned}$$

Feasibility of the heat exchange requires at the in- and outlet of the heat exchanger that

$$\begin{aligned} t_{i,k} &\geq t_{j,k} - \Gamma(1 - z_{ijk}), \quad \forall i \in H, k \in K, \\ t_{i,k+1} &\geq t_{j,k+1} - \Gamma(1 - z_{ijk}), \quad \forall i \in H, k \in K, \end{aligned}$$

Furthermore, no temperature cross-over may occur in the heat exchanger. Following the mathematical description developed in Section C.1, this can be guaranteed by

$$\Delta Q_{ijk}(T^*) \equiv \int_{t_{j,k+1}}^{T^*} F_j c_{p,j}(T)\, dT - \int_{t_{i,k+1}}^{T^*+\Delta T_{\min}} F_i c_{p,i}(T)\, dT \geq -\Omega(1 - z_{ijk}), \quad \forall T^* \in [t_{j,k+1}, t_{i,k}],$$

which is a generalized semi-infinite constraint since the set depends on the variables $t_{j,k+1}$ and $t_{i,k}$.

Lastly, it will be necessary to compute required areas for heat exchange. Suppose that the area can be calculated as $A = \frac{Q}{U\Delta T}$ where U is the heat transfer coefficient and log mean temperature difference

$$\Delta T = \frac{(t_{\text{in},1} - t_{\text{out},2}) - (t_{\text{out},1} - t_{\text{in},2})}{\ln \frac{t_{\text{in},1} - t_{\text{out},2}}{t_{\text{out},1} - t_{\text{in},2}}}.$$

Thus, one can calculate the necessary areas A_{ijk}, Acu_i, and Ahu_j.

The objective can be written as

$$\begin{aligned} \min \sum_{i \in H} &(\text{ccu}\, qcu_i + \text{cfcu}_i\, zcu_i + \text{cacu}_i\, Acu_i) \\ &+ \sum_{j \in C} (\text{chu}\, qhu_j + \text{cfhu}_j\, zhu_j + \text{cahu}_j\, Ahu_j) \\ &+ \sum_{i \in H} \sum_{j \in C} \sum_{k \in K} (\text{cf}_{ij}\, z_{ijk} + \text{ca}_{ij}\, A_{ijk}) \end{aligned}$$

where ccu and chu are the unit costs of cold and hot utility, respectively, cfcu_i, cfhu_j, and cf_{ij} are the respective fixed costs for each heat exchanger and cacu_i, cahu_j, and ca_{ij} are the respective area dependent costs for each heat exchanger.

Bibliography

[1] C. S. Adjiman and C. A. Floudas. Rigorous convex underestimators for general twice-differentiable problems. *Journal of Global Optimization*, 9:23–40, 1996.

[2] 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.

[3] G. Alefeld and G. Mayer. Interval analysis: theory and applications. *Journal of Computational and Applied Mathematics*, 121:421–464, Sept. 2000.

[4] R. Anantharaman, O. S. Abbas, and T. Gundersen. Energy level composite curves—a new graphical methodology for the integration of energy intensive processes. *Applied Thermal Engineering*, 26:1378–1384, Sept. 2006.

[5] 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.

[6] A. Aspelund and T. Gundersen. A new process synthesis methodology utilizing pressure exergy in subambient processes. In V. Plesu and P. S. Agachi, editors, *17th European Symposium on Computer Aided Process Engineering – ESCAPE17*, pages T5–623—T5–630. Elsevier, 2007.

[7] A. Aspelund and T. Gundersen. A liquefied energy chain for transport and utilization of natural gas for power production with CO2 capture and storage – Part 1. *Applied Energy*, 86:781–792, June 2009.

[8] A. Aspelund and T. Gundersen. A liquefied energy chain for transport and utilization of natural gas for power production with CO2 capture and storage – Part 2: The offshore and the onshore processes. *Applied Energy*, 86:793–804, June 2009.

[9] A. Aspelund, D. O. Berstad, and T. Gundersen. An extended pinch analysis and design procedure utilizing pressure based exergy for subambient cooling. *Applied Thermal Engineering*, 27:2633–2649, 2007.

[10] M. J. Bagajewicz, R. Pham, and V. Manousiouthakis. On the state space approach to mass/heat exchanger network design. *Chemical Engineering Science*, 53:2595–2621, 1998.

[11] S. Balendra and I. D. L. Bogle. Modular global optimisation in chemical engineering. *Journal of Global Optimization*, 45:169–185, 2009.

[12] A. Barbaro and M. J. Bagajewicz. New rigorous one-step MILP formulation for heat exchanger network synthesis. *Computers & Chemical Engineering*, 29(9):1945–1976, Aug. 2005.

[13] R. Barták. Theory and practice of constraint propagation. In *Proceedings of the 3rd Workshop on Constraint Programming in Decision and Control*, pages 7–14, 2001.

[14] P. I. Barton, R. J. Allgor, W. F. Feehery, and S. Galán. Dynamic optimization in a discontinuous world. *Industrial & Engineering Chemistry Research*, 37(3):966–981, 1998.

[15] V. D. Batukhtin. On solving discontinuous extremal problems. *Journal of Optimization Theory and Applications*, 77:575–589, June 1993.

[16] V. D. Batukhtin. An approach to the solution of discontinuous extremal problems. *Journal of Computer and Systems Sciences International*, 33:30–38, 1995.

[17] V. D. Batukhtin, S. I. Bigil'deev, and T. B. Bigil'deeva. Numerical methods for solutions of discontinuous extremal problems. *Journal of Computer and Systems Sciences International*, 36:438–445, 1997.

[18] E. Baumann. Optimal centered forms. *BIT Numerical Mathematics*, 28(1):80–87, Mar. 1988.

[19] B. T. Baumrucker and L. T. Biegler. MPEC strategies for optimization of a class of hybrid dynamic systems. *Journal of Process Control*, 19(8):1248–1256, Sept. 2009.

[20] B. T. Baumrucker, J. G. Renfro, and L. T. Biegler. MPEC problem formulations and solution strategies with chemical engineering applications. *Computers & Chemical Engineering*, 32:2903–2913, 2008.

[21] M. Beckers, V. Mosenkis, and U. Naumann. Adjoint mode computation of subgradients for McCormick relaxations. In S. Forth, P. Hovland, E. Phipps, J. Utke, and A. Walther, editors, *Recent Advances in Algorithmic Differentiation*, volume 87 of *Lecture Notes in Computational Science and Engineering*, pages 103–113. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[22] R. E. Bellman. *Dynamic programming*. Princeton University Press, 1957.

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

[24] P. Belotti, S. Cafieri, J. Lee, and L. Liberti. Feasibility-based bounds tightening via fixed points. In W. Wu and O. Daescu, editors, *Combinatorial Optimization and Applications*, volume 6508, pages 65–76, Berlin Heidelberg, 2010. Springer.

[25] F. Benhamou and W. J. Older. Applying interval arithmetic to real, integer, and boolean constraints. *The Journal of Logic Programming*, 32(1):1–24, July 1997.

[26] F. Benhamou, D. McAllester, and P. Van Hentenryck. CLP(intervals) revisited. In M. Bruynooghe, editor, *Proceedings of the 1994 International Symposium on Logic programming*, pages 124–138, Cambridge, MA, 1994. MIT Press.

[27] F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget. Revising hull and box consistency. In *Proceedings of the International Conference on Logic Programming, ICLP'99*, pages 230–244, Cambridge, MA, 1999. MIT Press.

[28] F. Benhamou, L. Granvilliers, and F. Goualard. Interval constraints: Results and perspectives. In *New trends in constraints*, volume 1865 of *Lecture Notes in Artificial Intelligence*, pages 1–16, Berlin, 2000. Springer.

[29] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, second edition, 2003.

[30] D. Bertsimas and J. N. Tsitsiklis. *Introduction to linear optimization*. Athena Scientific, Belmont, MA, 1997.

[31] C. Bessiere. Constraint propagation. In F. Rossi, P. van Beek, and T. Walsh, editors, *Handbook of Constraint Programming*, chapter 3, pages 29–83. Elsevier, Amsterdam, Netherlands, 2006.

[32] B. Bhattacharjee, W. H. Green, and P. I. Barton. Interval methods for semi-infinite programs. *Computational Optimization and Applications*, 30(1):63–93, Jan. 2005.

[33] L. T. Biegler, I. E. Grossmann, and A. W. Westerberg. *Systematic methods of chemical process design*. Prentice Hall PTR, Upper Saddle River, NJ, 1997.

[34] A. Bompadre and A. Mitsos. Convergence rate of McCormick relaxations. *Journal of Global Optimization*, 52:1–28, 2012.

[35] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[36] R. P. Byrne and I. D. L. Bogle. Global optimisation of constrained non-convex programs using reformulation and interval analysis. *Computers & Chemical Engineering*, 23:1341–1350, Nov. 1999.

[37] R. P. Byrne and I. D. L. Bogle. Global optimization of modular process flowsheets. *Industrial & Engineering Chemistry Research*, 39:4296–4301, Nov. 2000.

[38] A. Caprara and M. Locatelli. Global optimization problems and domain reduction strategies. *Mathematical Programming*, 125(1):123–137, Jan. 2010.

[39] M. Castier and E. M. Queiroz. Energy targeting in heat exchanger network synthesis using rigorous physical property calculations. *Industrial & Engineering Chemistry Research*, 41(6):1511–1515, Mar. 2002.

[40] B. Chachuat. MC++ - A versatile library for McCormick relaxations and Taylor models. `http://www3.imperial.ac.uk/people/b.chachuat/research/`, 2011.

[41] B. Chachuat, A. Mitsos, and P. I. Barton. libMC - A numeric library for McCormick relaxation of factorable functions. `http://yoric.mit.edu/libMC/`, 2007.

[42] J. J. J. Chen. Comments on improvements on a replacement for the logarithmic mean. *Chemical Engineering Science*, 42(10):2488–2489, 1987.

[43] N. Christofides. *Graph Theory: An Algorithmic Approach*. Academic Press, New York, NY, 1975.

[44] A. R. Ciric and C. A. Floudas. Heat exchanger network synthesis without decomposition. *Computers & Chemical Engineering*, 15:385–396, June 1991.

[45] F. H. Clarke. *Optimization and nonsmooth analysis*. Wiley, New York, NY, 1983.

[46] J. G. Cleary. Logical arithmetic. *Future Computing Systems*, 2(2):124–149, 1987.

[47] A. R. Conn and M. Mongeau. Discontinuous piecewise linear optimization. *Mathematical Programming*, 80(3):315–380, Feb. 1998.

[48] C. Corbett, M. Maier, M. Beckers, U. Naumann, A. Ghobeity, and A. Mitsos. Compiler-generated subgradient code for McCormick relaxations. Technical report, Department of Computer Science, Aachen, Germany, 2011.

[49] J. Cortés. Discontinuous dynamical systems. *IEEE Control Systems Magazine*, 28(3): 36–73, June 2008.

[50] T. Csendes and D. Ratz. Subdivision direction selection in interval methods for global optimization. *SIAM Journal on Numerical Analysis*, 34(3):922–938, 1997.

[51] E. Davis. Constraint propagation with interval labels. *Artificial Intelligence*, 32(3): 281–331, July 1987.

[52] L. C. W. Dixon and G. P. Szego. The optimization problem: An introduction. In L. C. W. Dixon and G. P. Szego, editors, *Towards Global Optimization*. North Holland, New York, NY, 1978.

[53] F. Domes and A. Neumaier. Constraint propagation on quadratic constraints. *Constraints*, 15(3):404–429, Aug. 2010.

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

[55] M. A. Duran and I. E. Grossmann. Simultaneous optimization and heat integration of chemical processes. *AIChE Journal*, 32:123–138, 1986.

[56] T. G. W. Epperly and E. N. Pistikopoulos. A reduced space branch and bound algorithm for global optimization. *Journal of Global Optimization*, 11(3):287–311, 1997.

[57] Y. M. Ermoliev and V. I. Norkin. On constrained discontinuous optimization. In *Proceedings of 3rd GAMM/IFIP Workshop, Stochastic optimization: Numerical methods and technical applications*, volume 458 of *Lecture Notes in Economics and Mathematical Systems*, pages 128–142, Berlin, 1998. Springer.

[58] Y. M. Ermoliev, V. I. Norkin, and R. J.-B. Wets. The minimization of semicontinuous functions: Mollifier subgradients. *SIAM Journal on Control and Optimization*, 33: 149–167, 1995.

[59] J. E. Falk and R. M. Soland. An algorithm for separable nonconvex programming problems. *Management Science*, 15:550–569, May 1969.

[60] X. Feng and X. X. Zhu. Combining pinch and exergy analysis process modifications. *Applied Thermal Engineering*, 17:249–261, 1997.

[61] M. C. Ferris, S. P. Dirkse, J.-H. Jagla, and A. Meeraus. An extended mathematical programming framework. *Computers & Chemical Engineering*, 33(12):1973–1982, 2009.

[62] C. A. Floudas, A. R. Ciric, and I. E. Grossmann. Automatic synthesis of optimum heat exchanger network configurations. *AIChE Journal*, 32:276–290, Feb. 1986.

[63] R. Fourer and D. Orban. Drampl: a meta solver for optimization problem analysis. *Computational Management Science*, 7(4):437–463, Aug. 2009.

[64] R. Fourer, C. Maheshwari, A. Neumaier, D. Orban, and H. Schichl. Convexity and concavity detection in computational graphs: Tree walks for convexity assessment. *INFORMS Journal on Computing*, pages 1–18, July 2009.

[65] K. C. Furman and N. V. Sahinidis. A critical review and annotated bibliography for heat exchanger network synthesis in the 20th century. *Industrial & Engineering Chemistry Research*, 41:2335–2370, May 2002.

[66] D. M. Gay. Computing perturbation bounds for nonlinear algebraic equations. *SIAM Journal on Numerical Analysis*, 20(3):638–651, June 1983.

[67] R. Goebel, R. G. Sanfelice, and A. R. Teel. Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2):28–93, Apr. 2009.

[68] R. A. Gordon. *The integrals of Lebesgue, Denjoy, Perron, and Henstock*. American Mathematical Society, Providence, RI, 1994.

[69] L. Granvilliers and F. Benhamou. Algorithm 852: RealPaver: An interval solver using constraint satisfaction techniques. *ACM Transactions on Mathematical Software*, 32(1): 138–156, Mar. 2006.

[70] A. Griewank and A. Walther. *Evaluating Derivatives*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, Jan. 2008.

[71] I. E. Grossmann, H. Yeomans, and Z. Kravanja. A rigorous disjunctive optimization model for simultaneous flowsheet optimization and heat integration. *Computers & Chemical Engineering*, 22:S157–S164, 1998.

[72] T. Gundersen and L. Naess. The synthesis of cost optimal heat exchanger networks: An industrial review of the state of the art. *Computers & Chemical Engineering*, 12: 503–530, 1988.

[73] E. Hansen and S. Sengupta. Bounding solutions of systems of equations using interval analysis. *BIT*, 21(2):203–211, June 1981.

[74] E. Hansen and G. W. Walster. *Global optimization using interval analysis*. Marcel Dekker, Inc., New York, NY, 2nd edition, 2004.

[75] P. Hansen, B. Jaumard, and S.-H. Lu. An analytical approach to global optimization. *Mathematical Programming*, 52(1-3):227–254, May 1991.

[76] M. M. F. Hasan, I. A. Karimi, and H. E. Alfadala. Synthesis of heat exchanger networks involving phase changes. In H. Alfadala, M. M. El-Halwagi, and G. R. Reklaitis, editors, *1st Annual Gas Processing Symposium*, pages 1–8. Elsevier, 2009.

[77] M. M. F. Hasan, G. Jayaraman, I. A. Karimi, and H. E. Alfadala. Synthesis of heat exchanger networks with nonisothermal phase changes. *AIChE Journal*, 56:930–945, 2010.

[78] C. A. Haverly. Studies of the behavior of recursion for the pooling problem. *ACM SIGMAP Bulletin*, 25:19–28, Dec. 1978.

[79] B. Hayes. An adventure in the Nth dimension. *American Scientist*, 99(6):442–446, 2011.

[80] P. R. Heyl. Properties of the locus r=constant in the space of n dimensions. In *Publications of the University of Pennsylvania*, chapter 2, pages 33–39. University of Pennsylvania, Philadelphia, PA, 1897. URL `http://books.google.com/books?id=j5pQAAAAYAAJ`.

[81] T. H. Hildebrandt and L. M. Graves. Implicit functions and their differentials in general analysis. *Transactions of the American Mathematical Society*, 29(1):127–153, Jan. 1927.

[82] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer, Berlin, 1993.

[83] E. C. Hohmann. *Optimal networks for heat exchange*. Ph.D. thesis, University of Southern California, Los Angeles, CA, 1971.

[84] K. Holiastos and V. Manousiouthakis. Minimum hot/cold/electric utility cost for heat exchange networks. *Computers & Chemical Engineering*, 26:3–16, 2002.

[85] M. Homšak and P. Glavič. Pressure exchangers in pinch technology. *Computers & Chemical Engineering*, 20(6–7):711–715, July 1996.

[86] J. Hooker. *Logic-based Methods for Optimization: Combining optimization and constraint satisfaction*. John Wiley & Sons, Inc., New York, NY, 2000.

[87] R. Horst. Deterministic global optimization with partition sets whose feasibility is not known: Application to concave minimization, reverse convex constraints, DC-programming, and Lipschitzian optimization. *Journal of Optimization Theory and Applications*, 58(1):11–37, July 1988.

[88] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer, Berlin, third edition, 1996.

[89] E. Hyvönen. Constraint reasoning based on interval arithmetic: the tolerance propagation approach. *Artificial Intelligence*, 58(1-3):71–112, Dec. 1992.

[90] L. Jaulin. Solving set-valued constraint satisfaction problems. *Computing*, 94(2-4): 297–311, Nov. 2012.

[91] L. Jaulin, K. Michel, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer, London, UK, 2001.

[92] J. Jezowski. Heat exchanger network grassroot and retrofit design. the review of the state-of-the art: Part I. Heat exchanger network targeting and insight based methods of synthesis. *Hungarian Journal of Industrial Chemistry*, 22:279–294, 1994.

[93] J. Jezowski. Heat exchanger network grassroot and retrofit design. the review of the state-of-the art: Part II. Heat exchanger network synthesis by mathematical methods and approaches for retrofit design. *Hungarian Journal of Industrial Chemistry*, 22: 295–308, 1994.

[94] R. B. Kearfott and S. Hongthong. Validated linear relaxations and preprocessing: Some experiments. *SIAM Journal on Optimization*, 16:418–433, 2005.

[95] R. B. Kearfott, M. Nakao, A. Neumaier, S. M. Rump, S. Shary, and P. Van Hentenryck. Standardized notation in interval analysis. In *Proc. XIII Baikal International School-seminar "Optimization methods and their applications"*, volume 4, pages 106–113, Irkutsk, Russia, 2005. Institute of Energy Systems SB RAS.

[96] R. B. Kearfott, J. Castille, and G. Tyagi. A general framework for convexity analysis in deterministic global optimization. *Journal of Global Optimization*, 56(3):765–785, June 2013.

[97] P. Kesavan, R. J. Allgor, E. P. Gatzke, and P. I. Barton. Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs. *Mathematical Programming*, 100:517–535, May 2004.

[98] K. C. Kiwiel. *Methods of descent for nondifferentiable optimization*. Springer, Berlin, 1985.

[99] O. Knüppel. PROFIL/BIAS–A fast interval library. *Computing*, 53(3-4):277–287, Sept. 1994.

[100] L. V. Kolev and I. P. Nenov. Cheap and tight bounds on the solution set of perturbed systems of nonlinear equations. *Reliable Computing*, 7(5):399–408, 2001. doi: 10.1023/A:1011475926711.

[101] T. J. Kotas. *The exergy method of thermal plant analysis*. Krieger Publishing Company, Malabar, FL, 1995.

[102] R. Krawczyk. Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken. *Computing*, 4(3):187–201, Sept. 1969.

[103] R. Krawczyk and A. Neumaier. Interval slopes for rational functions and associated centered forms. *SIAM Journal on Numerical Analysis*, 22(3):604–616, 1985.

[104] R. Krawczyk and K. Nickel. Die zentrische Form in der Intervallarithmetik, ihre quadratische Konvergenz und ihre Inklusionsisotonie. *Computing*, 28(2):117–137, June 1982.

[105] R. Krawczyk and F. Selsmark. Order-convergence and iterative interval methods. *Journal of Mathematical Analysis and Applications*, 73(1):1–23, Jan. 1980.

[106] Y. Lebbah, C. Michel, M. Rueher, D. Daney, and J.-P. Merlet. Efficient and safe global constraints for handling numerical constraint systems. *SIAM Journal on Numerical Analysis*, 42(5):2076, 2005.

[107] O. Lhomme. Consistency techniques for numeric CSPs. In *International Joint Conference on Artificial Intelligence*, pages 232–238, 1993.

[108] B. Linnhoff and J. R. Flower. Synthesis of heat exchanger networks: I. Systematic generation of energy optimal networks. *AIChE Journal*, 24:633–642, 1978.

[109] B. Linnhoff and J. R. Flower. Synthesis of heat exchanger networks: II. Evolutionary generation of networks with various criteria of optimality. *AIChE Journal*, 24:642–654, July 1978.

[110] B. Linnhoff and D. R. Vredeveld. Pinch technology has come of age. *Chemical Engineering Progress*, 80:33–40, 1984.

[111] B. Linnhoff, D. W. Townsend, D. Boland, G. F. Hewitt, B. E. A. Thomas, A. R. Guy, and R. H. Marsland. *A User Guide on Process Integration for the Efficient Use of Energy*. Institution of Chemical Engineers, Rugby, UK, second edition, 1992.

[112] J. Liu, L.-Z. Liao, A. Nerode, and J. H. Taylor. Optimal control of systems with continuous and discrete states. In *Proceedings of 32nd IEEE Conference on Decision and Control*, pages 2292–2297. IEEE, 1993.

[113] L. Lukšan and J. Vlček. Algorithm 811: NDA: Algorithms for nondifferentiable optimization. *ACM Transactions on Mathematical Software*, 27:193–213, 2001.

[114] A. K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1): 99–118, Feb. 1977.

[115] M. M. Mäkelä and P. Neittaanmäki. *Nonsmooth Optimization*. World Scientific, Singapore, 1992.

[116] C. D. Maranas and C. A. Floudas. Global minimum potential energy conformations of small molecules. *Journal of Global Optimization*, 4(2):135–170, 1994.

[117] G. Mayer. Epsilon-inflation in verification algorithms. *Journal of Computational and Applied Mathematics*, 60(1-2):147–169, June 1995.

[118] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I – Convex underestimating problems. *Mathematical Programming*, 10: 147–175, Dec. 1976.

[119] G. P. McCormick. *Nonlinear programming: theory, algorithms, and applications*. Wiley, New York, NY, 1983.

[120] G. Melquiond, S. Pion, and H. Brönnimann. Boost interval arithmetic library. `http://www.boost.org/doc/libs/1_49_0/`, 2006.

[121] A. Mitsos, B. Chachuat, and P. I. Barton. McCormick-based relaxations of algorithms. *SIAM Journal on Optimization*, 20:573–601, 2009.

[122] R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, PA, 1979.

[123] R. E. Moore and S. T. Jones. Safe starting regions for iterative methods. *SIAM Journal on Numerical Analysis*, 14(6):1051–1065, Dec. 1977.

[124] L. Moreau and D. Aeyels. Optimization of discontinuous functions: A generalized theory of differentiation. *SIAM Journal on Optimization*, 11:53–69, 2000.

[125] Y. Nesterov and A. Nemirovski. *Interior Point Polynomial Methods in Convex Programming*. SIAM, Philadelphia, PA, 1994.

[126] A. Neumaier. Rigorous sensitivity analysis for parameter-dependent systems of equations. *Journal of Mathematical Analysis and Applications*, 144(1):16–25, Nov. 1989.

[127] A. Neumaier. *Interval methods for systems of equations*. Cambridge University Press, Cambridge, UK, 1990.

[128] A. Neumaier. Taylor forms—use and limits. *Reliable Computing*, 9:43–79, 2003.

[129] A. Neumaier. Complete search in continuous global optimization and constraint satisfaction. In A. Iserles, editor, *Acta Numerica*, volume 13, pages 271–370. Cambridge University Press, Cambridge, UK, 2004.

[130] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 1999.

[131] S. A. Papoulias and I. E. Grossmann. A structural optimization approach in process synthesis—I. Utility systems. *Computers & Chemical Engineering*, 7:695–706, 1983.

[132] S. A. Papoulias and I. E. Grossmann. A structural optimization approach in process synthesis—II. Heat recovery networks. *Computers & Chemical Engineering*, 7:707–721, 1983.

[133] S. A. Papoulias and I. E. Grossmann. A structural optimization approach in process synthesis—III. Total processing systems. *Computers & Chemical Engineering*, 7:723–734, 1983.

[134] J. M. Ponce-Ortega, A. Jiménez-Gutiérrez, and I. E. Grossmann. Optimal synthesis of heat exchanger networks involving isothermal process streams. *Computers & Chemical Engineering*, 32:1918–1942, Aug. 2008.

[135] J. M. Prausnitz. Isentropic compression of nonideal gases. *Industrial & Engineering Chemistry*, 47:1032–1033, May 1955.

[136] R. E. Pugh. A language for nonlinear programming problems. *Mathematical Programming*, 2(1):176–206, Feb. 1972.

[137] H. Ratschek. Centered forms. *SIAM Journal on Numerical Analysis*, 17(5):656–662, 1980.

[138] H. Ratschek and J. Rokne. *Computer methods for the range of functions*. Ellis Horwood, Chichester, UK, 1984.

[139] D. Ratz. *Automatic Slope Computation and its Application in Nonsmooth Global Optimization*. Shaker, Aachen, Germany, 1998.

[140] D. Ratz and T. Csendes. On the selection of subdivision directions in interval branch-and-bound methods for global optimization. *Journal of Global Optimization*, 7(2): 183–207, Sept. 1995.

[141] R. C. Reid, J. M. Prausnitz, and B. E. Polling. *The properties of gases and liquids*. McGraw-Hill, New York, NY, fourth edition, 1987.

[142] F. N. Ris. *Interval analysis and applications to linear algebra*. Ph.D. thesis, University of Oxford, 1972.

[143] R. T. Rockafellar. *Convex analysis*. Princeton University Press, Princeton, NJ, 1996.

[144] R. Y. Rubinstein. Smoothed functionals in stochastic optimization. *Mathematics of Operations Research*, 8:26–33, Feb. 1983.

[145] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, NY, third edition, 1976.

[146] S. Rump. INTLAB - INTerval LABoratory. In T. Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. `http://www.ti3.tuhh.de/rump/`.

[147] S. M. Rump. Rigorous sensitivity analysis for systems of linear and nonlinear equations. *Mathematics of Computation*, 54(190):721–721, May 1990.

[148] 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, May 1995.

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

[150] N. V. Sahinidis. BARON solver manual. `http://gams.com/dd/docs/solvers/baron.pdf`, 2012.

[151] N. V. Sahinidis and M. Tawarmalani. BARON solver manual. `http://gams.com/dd/docs/solvers/baron.pdf`, 2009.

[152] D. Sam-Haroud and B. Faltings. Consistency techniques for continuous constraints. *Constraints*, 1(1-2):85–118, Sept. 1996.

[153] H. Schichl and A. Neumaier. Interval analysis on directed acyclic graphs for global optimization. *Journal of Global Optimization*, 33(4):541–562, Dec. 2005.

[154] A. Schöbel and D. Scholz. The theoretical and empirical rate of convergence for geometric branch-and-bound methods. *Journal of Global Optimization*, 48(3):473–495, Dec. 2010.

[155] J. K. Scott. *Reachability Analysis and Deterministic Global Optimization of Differential-Algebraic Systems*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2012.

[156] J. K. Scott, M. D. Stuber, and P. I. Barton. Generalized McCormick relaxations. *Journal of Global Optimization*, 51(4):569–606, Feb. 2011.

[157] W. D. Seider, D. R. Lewin, J. D. Seader, and S. Widagdo. *Process Design Principles: Synthesis, Analysis and Evaluation*. John Wiley & Sons, Inc., New York,, third edition, 2008.

[158] U. V. Shenoy. *Heat exchanger network synthesis: The pinch technology-based approach*. Gulf Publishing Company, Houston, TX, 1995.

[159] E. M. B. Smith and C. C. Pantelides. Global optimisation of nonconvex MINLPs. *Computers & Chemical Engineering*, 21:S791–S796, 1997.

[160] R. Smith. *Chemical Process Design and Integration*. John Wiley & Sons, Ltd, West Sussex, UK, 2005.

[161] J. Stolfi and L. H. de Figueiredo. *Self-Validated Numerical Methods and Applications*. Brazilian Mathematics Colloquium monographs. IMPA/CNPq, Rio de Janeiro, Brazil, 1997.

[162] M. D. Stuber. *Evaluation of Process Systems Operating Envelopes*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2013.

[163] M. D. Stuber and P. I. Barton. Robust simulation and design using semi-infinite programs with implicit functions. *International Journal of Reliability and Safety*, 5(3-4): 378–397, 2011.

[164] M. D. Stuber, J. K. Scott, and P. I. Barton. Global optimization of implicit functions, 2013. Submitted.

[165] M. Tawarmalani and N. V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming*. Kluwer Academic Publishers, Dordrecht, 2002.

[166] M. Tawarmalani and N. V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99: 563–591, Apr. 2004.

[167] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, May 2005.

[168] M. Türkay and I. E. Grossmann. Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs-multiple size regions. *Industrial & Engineering Chemistry Research*, 35:2611–2623, Jan. 1996.

[169] P. Van Hentenryck, D. McAllester, and D. Kapur. Solving polynomial systems using a branch and prune approach. *SIAM Journal on Numerical Analysis*, 34(2):797–827, Apr. 1997.

[170] P. Van Hentenryck, L. Michel, and F. Benhamou. Constraint programming over nonlinear constraints. *Science of Computer Programming*, 30(1-2):83–118, Jan. 1998.

[171] R. J. Van Iwaarden. *An improved unconstrained global optimization algorithm*. Ph.D. thesis, University of Colorado at Denver, Denver, CO, 1996.

[172] L. N. Vicente and A. L. Custódio. Analysis of direct searches for discontinuous functions. *Mathematical Programming*, 133(1-2):299–325, Dec. 2012.

[173] X.-H. Vu, D. Sam-Haroud, and M.-C. Silaghi. Numerical constraint satisfaction problems with non-isolated solutions. In C. Bliek, C. Jermann, and A. Neumaier, editors, *Global optimization and constraint satisfaction*, volume 2861 of *Lecture Notes in Computer Science*, pages 194–210, Berlin, 2003. Springer.

[174] X.-H. Vu, H. Schichl, and D. Sam-Haroud. Interval propagation and search on directed acyclic graphs for numerical constraint solving. *Journal of Global Optimization*, 45(4):499–531, Dec. 2009.

[175] W. Walter. *Analysis II*. Springer, Berlin, second edition, 1991.

[176] A. Wechsung and P. I. Barton. Global optimization of bounded factorable functions with discontinuities. *Journal of Global Optimization*, 2013. doi: 10.1007/s10898-013-0060-3. In press.

[177] A. Wechsung, A. Aspelund, T. Gundersen, and P. I. Barton. Synthesis of heat exchanger networks at subambient conditions with compression and expansion of process streams. *AIChE Journal*, 57(8):2090–2108, Aug. 2011.

[178] A. Wechsung, S. D. Schaber, and P. I. Barton. The cluster problem revisited. *Journal of Global Optimization*, 2013. doi: 10.1007/s10898-013-0059-9. In press.

[179] T. F. Yee and I. E. Grossmann. Simultaneous optimization models for heat integration—II. Heat exchanger network synthesis. *Computers & Chemical Engineering*, 14:1151–1164, 1990.

[180] T. F. Yee, I. E. Grossmann, and Z. Kravanja. Simultaneous optimization models for heat integration—I. Area and energy targeting and modeling of multi-stream exchangers. *Computers & Chemical Engineering*, 14:1165–1184, 1990.

[181] T. F. Yee, I. E. Grossmann, and Z. Kravanja. Simultaneous optimization models for heat integration—III. Process and heat exchanger network optimization. *Computers & Chemical Engineering*, 14:1185–1200, 1990.

[182] I. Zang. Discontinuous optimization by smoothing. *Mathematics of Operations Research*, 6:140–152, Feb. 1981.

[183] Q. Zheng. Robust analysis and global minimization of a class of discontinuous functions (I). *Acta Mathematicae Applicatae Sinica*, 6:205–223, July 1990.