DISS. ETH Nr. 21024

Convex Relaxations for Mixed-Integer Nonlinear Programs

ABHANDLUNG
zur Erlangung des Titels

DOKTOR DER WISSENSCHAFTEN
der
ETH Zürich

vorgelegt von

MARTIN BALLERSTEIN

Dipl.-Wirt.-Math., Otto-von-Guericke Universität Magdeburg
geboren am 30. Oktober 1983 in Zerbst
Deutscher Staatsbürger

Angenommen auf Antrag von
Prof. Dr. Robert Weismantel
Prof. Dr. Alexander Martin

2013

Acknowledgments

Many people are to be acknowledged for their support in my endeavor to write this thesis. First of all, I wish to express my deepest gratitude to Robert Weismantel for the continuous discussions about mathematical optimization and life at all. Thanks for the opportunity to work in such an enthusiastic team of researchers both at the Institute of Mathematical Optimization at the Otto-von-Guericke University Magdeburg and the Institute for Operations Research at ETH Zurich.

I am thankful to Alexander Martin for accepting to be the co-examiner of this thesis.

I am particularly grateful to Dennis Michaels for his continuous support and supervision of this thesis. After writing my diploma thesis under his supervision, I had the chance to continue the started project in my doctoral studies. Based on many fruitful discussions and his mathematical intuition, I could not only finish this project but also start further research on novel and interesting topics. He steadily motivated me to get to the heart of the problems. Thanks a lot.

This thesis is a compilation of five joint projects in each of which I benefited from the comments and ideas of my colleagues. I thank Dennis Michaels, Andreas Seidel-Morgenstern, and Robert Weismantel for the joint work on the paper [BMSMW10]. For the work [BKK+11] and the submitted extension [BKK+] I owe a debt of gratitude to Achim Kienle, Christian Kunde, Dennis Michaels, and Robert Weismantel. I thank Dennis Michaels and Stefan Vigerske for the joint implementation project which is documented and evaluated in the technical report [BMV13]. In particular, I appreciate the lessons I learned from Stefan Vigerske regard-

ing coding. Many thanks to Dennis Michaels for the collaboration on the accepted paper [BM]. Finally, I thank Dennis Michaels and Robert Weismantel for the discussions about the material of the last chapter of this thesis.

I am grateful to the German Research Foundation (DFG) which supported parts of this work through the grant FOR 468 and the Collaborative Research Centre "Integrated Chemical Processes in Liquid Multiphase Systems" (CRC/Transregio 63 "InPROMPT"). Moreover, I was supported by the Research Focus "Dynamical Systems in Biomedicine and Process Engineering" of the Ministry of Education Saxony-Anhalt, Germany. I further thank the people at the International Max Planck Research School Magdeburg, Germany.

Special thanks go to my colleagues both at IMO in Magdeburg and at IFOR in Zurich. I enjoyed jogging with a lot of you, tasting and promoting Feuerzangenbowle before Christmas, several lessons in unihockey and table tennis, and all the other little things which make life beautiful. Especially, I would like to thank Utz-Uwe Haus for his support regarding the computational environment and my office mates Matthias Jach, Raymond Hemmecke, Luis Torres, Kent Andersen, Dennis Michaels, and Christian Wagner for their willingness to help me with all the little problems.

Martin Ballerstein, Zurich, September 2013

Zusammenfassung

Die vorliegende Dissertationsschrift beschäftigt sich mit neuen Techniken zur Erzeugung von starken konvexen Relaxierungen für gemischt-ganzzahlige nichtlineare Optimierungsprobleme (MINLP). Während lokale Optimierungssoftware sehr schnell vielversprechende Betriebspunkte eines MINLPs bestimmen kann, liefert die Lösung der konvexen Relaxierung eine globale Schranke für das MINLP, die dafür genutzt werden kann, die Qualität der lokalen Lösung zu bewerten. Die Effizienz dieses Bewertungsansatzes ist natürlich stark beeinflusst von der Stärke der konvexen Relaxierung.

Konvexe Relaxierungen von allgemeinen MINLPs werden dadurch erzeugt, dass jede nichtlineare Funktion in der Modellbeschreibung durch konvexe unter- und konkave überschätzende Funktionen ersetzt wird. In diesem Zusammenhang ist es wünschenswert, immer die bestmöglichen konvexen Unter- und konkaven Überschätzer einer Funktion über einem vorgegebenen Definitionsbereich zu verwenden - die sogenannten konvexen beziehungsweise konkaven Einhüllenden. Die Berechnung dieser Einhüllenden kann allerdings sehr schwierig sein, so dass analytische Ausdrücke nur für einige Klassen von wohl strukturierten Funktionen bekannt sind.

Ein anderer Faktor, der die Stärke der Unter- und Überschätzer beeinflusst, ist die Größe des zugrunde liegenden Definitionsbereichs: Je kleiner der Definitionsbereich, desto stärker sind die Unter- und Überschätzer. In vielen Anwendungen werden die Definitionsbereiche allerdings zu konservativ gewählt, während kleinere Bereiche implizit durch die Nebenbedingungen des MINLPs gegeben sind. Daher sind Techniken

zur Verkleinerung des Definitionsbereichs, welche auf der Analyse der Nebenbedingungen basieren, von entscheidender Bedeutung um Unter- und Überschätzer zu verbessern und um globale Optimierungsalgorithmen zu beschleunigen.

Der Schwerpunkt dieser Dissertationsschrift liegt auf der Entwicklung und rechnergestützten Analyse neuer konvexer Relaxierungen für MINLPs, insbesondere für zwei Anwendungen aus der Verfahrenstechnik. Hierbei handelt es sich einerseits um eine neue Technik zur Verkleinerung des Definitionsbereichs für eine allgemeine Struktur, die zur Modellierung chemischer Prozesse genutzt wird. Anderseits werden unterschiedliche Ansätze zur Erzeugung starker konvexer Relaxierungen für verschiedenste nichtlineare Funktionen präsentiert.

Zunächst liegt der Fokus auf der Bestimmung eines optimalen Designs für hybride Destillations- und Schmelzkristallisationsprozesse, das heißt einer neuartigen Prozesskonfiguration, die zur Auftrennung eines Stoffgemisches genutzt wird. Für die mathematische Beschreibung sowohl dieses Prozesses als auch anderer Separierungsprozesse ist es entscheidend die Massenerhaltung innerhalb des Prozesses zu modellieren. Basierend auf den analytischen Eigenschaften des entsprechenden Gleichungssystems wird eine Technik zur Verkleinerung des Definitionsbereichs der dazugehörigen Variablen vorgestellt. Die Anwendung dieser Technik ermöglicht es im Vergleich zu Standardsoftware, die Berechnung globaler Lösungen von hybriden Destillations- und Schmelzkristallisationsprozessen signifikant zu beschleunigen.

Danach liegt das Hauptaugenmerk der Arbeit auf der Erzeugung von konvexen Relaxierungen für nichtlineare Funktionen. Als Erstes werden bereits vorhandene Ergebnisse für zwei Klassen von interessanten, bivariaten Funktionen genutzt. Zum einen wird ein Schnittebenenalgorithmus ausgearbeitet, implementiert und analysiert, der sich für bivariate Funktionen eignet, die konvex oder konkav in jeder Variable sind und bei denen das Vorzeichen der Determinante der Hesse-Matrix über dem gesamten Definitionsbereich immer das gleiche ist. Zum anderen werden Relaxierungsstrategien für fortgeschrittene Gleichgewichtsfunktionen in chromatographischen Separierungsprozessen untersucht und angewendet, um die zulässigen Trennregionen dieser Prozesse komplett zu beschreiben.

Als Zweites wird vorgeschlagen die konvexen Einhüllenden in einem erweiterten Raum herzuleiten, um die kombinatorischen Schwierigkeiten zu überwinden, die sich bei der Berechnung der Einhüllenden im Origi-

nalraum ergeben. Insbesondere wird eine Klasse von Funktionen betrachtet, die einen Großteil aller nichtlinearen Funktionen in häufig verwendeten Problembibliotheken ausmacht. Diese Funktionen sind komponentenweise konkav in einem Teil der Variablen und konvex im anderen Teil. Für diese allgemeine Klasse von Funktionen sind die konvexen Einhüllenden bisher nicht bekannt. In dieser Arbeit werden explizite Formeln für eine erweiterte Formulierung der konvexen Einhüllenden dieser Funktionen hergeleitet, basierend auf einer simultanen Konvexifizierung mit multilinearen Monomen. Durch diese Herleitung wird nicht nur eine erweiterte Formulierung der konvexen Einhüllenden bestimmt, sondern auch eine starke simultane Relaxierung der Funktion und der multilinearen Monome. Etliche Beispiele zeigen, dass die simultane Relaxierung um Größenordnungen besser sein kann als die individuelle Relaxierung der Funktionen.

Inspiriert durch die Stärke und den rechentechnischen Einfluss der simultanen Relaxierung einer Funktion und multilinearer Monome wird abschließend die simultane Relaxierung von mehreren Funktionen in einem allgemeinen Kontext behandelt. Solch ein simultaner Ansatz erlaubt eine bedeutend bessere Relaxierung eines MINLPs, dessen Formulierung mehrere Funktionen in den gleichen Variablen beinhaltet, da die gegenseitigen Abhängigkeiten zwischen den verschiedenen Funktionen berücksichtigt werden. Dafür wird die simultane konvexe Hülle verschiedener Funktionen studiert und es werden theoretische Resultate bezüglich ihrer inneren und äußeren Darstellung mithilfe der Theorie der konvexen Einhüllende hergeleitet. Weiterhin werden diese Resultate ausgenutzt, um geschlossene Formeln für starke konvexe Relaxierungen von mehreren univariaten konvexen Funktionen abzuleiten.

Für jede Konvexifizierungstechnik sind Implementierungen verfügbar, die als Plugins für die open-source MINLP-Software SCIP genutzt werden können. Die Rechenergebnisse verschiedenster Fallbeispiele demonstrieren den Nutzen der vorgeschlagenen Techniken im Vergleich zu den heute genutzten Methoden.

Summary

This thesis deals with new techniques to construct a strong convex relaxation for a mixed-integer nonlinear program (MINLP). While local optimization software can quickly identify promising operating points of MINLPs, the solution of the convex relaxation provides a global bound on the optimal value of the MINLP that can be used to evaluate the quality of the local solution. Certainly, the efficiency of this evaluation is strongly dependent on the quality of the convex relaxation.

Convex relaxations of general MINLPs can be constructed by replacing each nonlinear function occurring in the model description by convex underestimating and concave overestimating functions. In this setting, it is desired to use the best possible convex underestimator and concave overestimator of a given function over an underlying domain – the so-called convex and concave envelope, respectively. However, the computation of these envelopes can be extremely difficult so that analytical expressions for envelopes are only available for some classes of well-structured functions.

Another factor influencing the strength of the estimators is the size of the underlying domain: The smaller the domain, the better the quality of the estimators. In many applications the initial domains of the variables are chosen rather conservatively while tighter bounds are implicitly given by the constraint set of the MINLP. Thus, bound tightening techniques, which exploit the information of the constraint set, are an essential ingredient to improve the estimators and to accelerate global optimization algorithms.

The focus of this thesis lies on the development and computational analysis of new convex relaxations for MINLPs, especially for two applications from chemical engineering. In detail, we derive a new bound tightening technique for a general structure used for modeling chemical processes and provide different approaches to generate strong convex relaxations for various nonlinear functions.

Initially, we aim at the optimal design of hybrid distillation/melt-crystallization processes, a novel process configuration to separate a m ixture into its component. A crucial part in the formal representation of this process as well as other separation processes is to model the mass conservation within the process. We exploit the analytical properties of the corresponding equation system to reduce the domains of the involved variables. Using the proposed technique, we can accelerate the computations for hybrid distillation/melt-crystallization processes significantly compared to standard software.

Then, we concentrate on the generation of convex relaxations for nonlinear functions. First, we exploit the existing theory for two interesting classes of bivariate functions. On the one hand, we elaborate, implement, and illustrate the strength of a cut-generation algorithm for bivariate functions which are convex or concave in each variable and for which the sign of the Hessian is the same over the entire domain. On the other hand, relaxation strategies for advanced equilibrium functions in chromatographic separation processes are analyzed and finally applied to completely describe the feasible separation regions of these processes.

Second, we suggest to derive the envelopes in an extended space to overcome the combinatorial difficulties involved in the computation of the convex envelope in the original space. In particular, we consider a class of functions accounting for a large amount of all nonlinearities in common benchmark libraries. These functions are component-wise concave in one part of the variables and convex in the other part of the variables. For this general class of functions the convex envelopes in the original variable space have not been discovered so far. We provide closed-form expressions for the extended formulation of their convex envelopes based on the simultaneous convexification with multilinear monomials. By construction, this approach does not only yield an extended formulation for the convex envelope of a function, but also a strong simultaneous relaxation of the function and the involved multilinear monomials. Several examples show that this simultaneous relaxation can be orders of magnitude better than the individual relaxation of the functions.

Finally, inspired by the strength and the computational impact of the simultaneous relaxation of a function and multilinear monomials, we further focus on the simultaneous convexification of several functions. In such an approach the relaxation of a MINLP involving several functions in the same variables is much tighter because the interdependence between the different functions is taken into account. We study the simultaneous convex hull of several functions for which we derive theoretical results concerning their inner and outer description by means of the rich theory of convex envelopes. Moreover, we apply these results to provide formulas for tight convex relaxations of several univariate convex functions.

Implementations of all convexification techniques are available as plugins for the open-source MINLP solver SCIP. The computational results of several case studies reveal the benefit of the proposed techniques compared to state-of-the-art methods.

Contents

CHAPTER 1

Introduction

This thesis presents new techniques to compute global optima of a mixed-integer nonlinear program (MINLP) which can be most generally described as

$$\begin{aligned} \min\ f_0(x,y) \quad \text{s.t.}\quad & f_i(x,y) \le 0,\ i = 1,\dots,m, \\ & (x,y) \in D = [l,u] \cap (\mathbf{R}^{n-d} \times \mathbf{Z}^d), \end{aligned} \tag{MINLP}$$

where f_0 and f_i, $i = 1,\dots,m$, are real-valued functions $\mathbf{R}^{n-d} \times \mathbf{R}^d \to \mathbf{R}$. This adaptable framework provides a modeling language for a wide range of topics and applications. On the one side, the nonlinearity of a MINLP enables one to reflect many real-world concepts which often cannot be described in a linear way. On the other side, the mixture of discrete and continuous variables meets the demand of the growing complexity of decision processes. In this way, structural and operational variables as well as decision variables are integrated into one model. Practical problems which can be formulated as MINLP are, for instance, the design of networks, trim-loss in the paper-industry, airplane boarding, production planning, and facility location (cf. [BL12]). Further applications can be found in chemical engineering (cf. [Flo95]).

Due to the expressive power of MINLPs it is not surprising that there is a lack of computational methods to efficiently solve general MINLPs to global optimality. A common tool to approach these problems are deterministic algorithms (cf. [HT96]) whose two main components are local optimization solvers and efficient algorithms to construct and solve

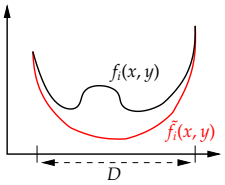
convex relaxations. While the local solvers determine a feasible solution whose objective function value constitutes an upper bound on the problem, the solution of the convex relaxation corresponds to a lower bound. Consequently, an optimal solution is found if the two bounds coincide.

The most popular deterministic algorithmic framework to solve MINLPs is the *branch-and-bound* algorithm, which successively subdivides the original problem into smaller subproblems until these subproblems can be solved to global optimality (cf. [BL12, Vig12] and references therein). The first step in this algorithm is to construct a convex relaxation over the initial domain. Based on the solution of this relaxation, the domain is divided into two subdomains and over each subdomain the same procedure is applied again. This branching step results in a branching tree in which each subdomain represents a node. A node is removed from the tree in the bounding step if the relaxation over the corresponding subdomain is infeasible or its lower bound is greater than or equal to the best known objective function value. Further deterministic algorithms are the outer-approximation algorithms [DG86], Generalized Benders decomposition (cf. [Flo95]), and a combinatorial approach introduced in [GKH+06, HMSMW07, Mic07].

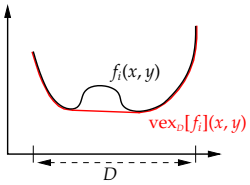
All deterministic algorithms have in common that their convergence heavily depends on the strength of the convex relaxations. For example, stronger relaxations allow to detect infeasibility of a node in the branch-and-bound algorithm more easily so that the exploration of further child nodes can be avoided. To construct strong relaxations, two main approaches are considered in this thesis, namely the *convex underestimation* of a function f_i over a given domain D and *bound tightening* techniques to infer smaller domains D from the constraint set $f_i(x, y) \leq 0$, $i = 1, \ldots, m$.

The use of convex underestimators for nonconvex functions f_i is a standard approach to construct a convex relaxation of a MINLP (cf. [BL12]). For this, each function f_i of the MINLP is replaced by a convex underestimating function $\tilde{f}_i(x, y)$ such that $\tilde{f}_i(x, y) \leq f_i(x, y)$ for all (x, y) of the current subdomain D. This is illustrated in Figure 1.1 (a), where a nonconvex function f_i is given in black while the convex underestimating function $\tilde{f}_i$ is depicted in red. In this setting the strength of the overall convex relaxation is defined via the strength of the individual convex underestimating functions: The stronger the underestimators, the stronger the relaxation. Therefore, it is desired to apply the best possible underestimator of a function f_i over a domain D – the so-called *convex envelope*, which is denoted by $\mathrm{vex}_D[f]$. The convex envelope of a function f_i is

displayed in Figure 1.1 (b) and its strength is apparent compared to the underestimator in Figure 1.1 (a).



(a) A convex underestimator.



(b) The best possible convex underestimator (convex envelope).

Figure 1.1.: Convex underestimators of a function $f(x, y)$ over D.

In general, the computation of the convex envelope is extremely hard so that closed-form expressions are only known for some classes of functions (cf. [KS12b]). A common technique to overcome this problem is to reformulate a given function f_i into sums and products of functions for which the convex envelope is known (cf. [McC76, TS04, BL12]).

Example 1.1. Consider the expression $f(x_1, x_2) = \frac{(x_1 x_2)^2}{1+\exp(x_1 x_2)}$. A typical way to reformulate this nonlinearity is to introduce four artificial variables $h_i \in \mathbf{R}$, $i = 1, 2, 3, 4$, and require

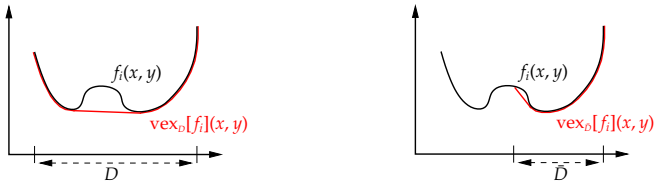
$$h_1 = h_2 h_3, \qquad h_2 = h_4^2, \qquad h_3 = \tfrac{1}{1+\exp(h_4)}, \qquad h_4 = x_1 x_2.$$

One can check that the functions h_2 and h_3 are convex over $\mathbf{R}_{\geq 0}$, i.e., the best convex underestimators are the functions itself. The best underestimator of the product terms h_1 and h_4 is known due to McCormick [McC76]. Thus, the composition of the underestimators yields an underestimator for the original function f. ◇

The underestimators generated by the reformulation technique are often not as strong as the convex envelope. As the speed of global optimization algorithms is closely related to the strength of the estimators, it is thus essential to derive further closed-form expressions for convex envelopes.

Besides the explicit formulas for convex underestimators, the size of the domain D is a crucial factor for the quality of a convex underestimator.

In Figures 1.2 (a) and (b) we illustrate the convex envelope of a function f_i over a larger and a smaller domain, respectively. The smaller domain in Figure 1.2 (b) leads to a significantly better underestimation of f_i over the concave part of f_i than the domain in Figure 1.2 (a).



(a) Large domain D.

(b) Smaller domain $\bar{D}$.

Figure 1.2.: Impact of the size of the domain on the relaxation quality of the convex envelope of a function f_i.

In many applications the domains of the variables are chosen rather conservatively and tighter bounds are implied by the constraint set $f_i(x, y) \leq 0$, $i = 1, \ldots, m$. For instance, the concentration variables in a separation process are assumed to be in the interval $[0, 1]$ while the purity requirements and the equations modeling such processes restrict the variables to be in much smaller intervals. This information is exploited by bound tightening techniques which take advantage of the constraint set to derive tighter bounds on the variables without losing any feasible or optimal solution. The huge impact of these reduction techniques on the quality of the relaxation made them an integral component of branch-and-bound algorithms which are thus often referred to as *branch-and-reduce* algorithms (cf. [RS96]).

Contributions and Structure of this Thesis This thesis deals with novel techniques to construct convex relaxations for MINLPs and their application in chemical engineering. In particular, we derive a new bound tightening technique for a general structure used for modeling chemical processes and provide various approaches to generate strong convex relaxations for nonconvex functions. For all techniques implementations are presented and their computational impact is demonstrated in several case studies, especially for two problems from chemical engineering.

In Chapter 2 we consider a sophisticated process configuration to separate a mixture into its components, namely hybrid distillation/melt-crystallization processes. Computationally, such processes are very challenging and even the global optimization of the distillation unit alone is still an open issue [GAB05]. To overcome this, we analyze the mathematical description of the distillation unit which mainly consists of *equilibrium equations* and *material balance equations*. Such equations model the course of the concentration variables within the distillation unit and are a general modeling tool for chemical processes.

We exploit the analytical properties of the equation system to propagate the high purity requirements, which specify the bounds on the concentrations of the products, through the distillation unit. This leads to a noticeable reduction of the domain and is used in two ways. On the one hand, the new bounds are applied in the original, highly nonlinear model to generate stronger relaxations. On the other hand, the course of the concentration variables is relaxed by the derived bounds such that the highly nonlinear system of equilibrium and material balance equations can be neglected. Although this approach only leads to a relaxed model formulation, it proves to be very efficient in order to determine infeasible or nonoptimal subdomains.

Furthermore, a comprehensive case study shows that the proposed techniques tremendously accelerate the computations of hybrid distillation/melt-crystallization processes compared to state-of-the-art software. We show some representative results in Table 1.1 which indicate that our proposed methods can enhance standard software by orders of magnitude. While the standard algorithms can only return lower bounds on the processes, our approach can prove global optimality after 6 minutes for the distillation unit and after 27 hours for the hybrid process.

	Optimal value	Lower bounds and CPU time by Standard software		Our approach	
Distillation	**306.3**	255.0	(100 hours)	**306.3**	(6 minutes)
Hybrid	**154.0**	57.2	(100 hours)	**154.0**	(27 hours)

Table 1.1.: Representative computational results of standard software and our approach for two chemical processes.

In Chapter 3 we utilize existing theory to compute strong underesti-

mators for two classes of interesting bivariate functions. Based on the work of Jach et al. [JMW08] we develop a cut-generation algorithm for the first class of bivariate functions exhibiting a *fixed convexity behavior*, i.e., the functions are convex or concave in each variable, and the sign of the determinant of the Hessian is the same over the entire domain. The authors provide a constructive procedure to determine the value of the convex envelope numerically which we exploit to construct supporting hyperplanes on the graph of the convex envelopes. The cut-generation algorithm is implemented in the open-source, mixed-integer nonlinear optimization solver SCIP [Ach07, Ach09] and is available in its standard distribution from version 2.1 onwards. Computational experiments reveal the strength of this new tool.

The second class of bivariate functions for which we investigate strong convex underestimators are *second-order isotherms*. They are a special type of equilibrium equations and are used to model the phase transition in chemical processes. Compared to conventional equilibrium models, second-order isotherms allow for more degrees of freedom in the modeling process so that certain chemical phenomena can be reflected. We analyze several reformulation strategies of the second-order isotherms into simpler functions for which the convex envelopes are known. Moreover, a lifting technique is proposed to derive tight underestimators without further reformulations and the additional introduction of artificial variables. The different underestimators are applied within the optimization of a *chromatographic separation process* so that not only the performance of the underestimators is evaluated, but further the behavior of chromatographic separation processes with second-order isotherms is completely described.

In Chapter 4 we continue the analysis of strong convex underestimators and, in particular, of convex envelopes. In contrast to the standard approach, we suggest to derive the convex envelope of a function f in an extended space based on the simultaneous convexification with multilinear monomials. The introduction of additional variables corresponding to the multilinear monomials allows to reduce the combinatorial difficulties involved in the analytical solution of the convex envelope. Although the additional variables can be seen as a disadvantage, this simultaneous relaxation can be orders of magnitude better compared to the individual relaxation of the monomials and f.

Example 1.2. Let $f(x) = {}^{x_1 x_2}/_{x_3}$, $x_1 \in [-1, 1], x_2 \in [0.1, 1], x_3 \in [0.1, 1]$. The

convex envelope of this function was derived in [KS12a]. In our setting we introduce additional variables z_{12}, z_{13}, z_{23}, and z_{123} for the monomials x_1x_2, x_1x_3, x_2x_3, and $x_1x_2x_3$, respectively, to compute the extended formulation of the convex envelope. In Table 1.2 we report the volumes of the individual convexification of the monomials and f, and the simultaneous convexification by the extended formulation. This difference in the volume accounts for a gap of 2120%. ◇

	Individual envelopes	Extended formulation
Volume	0.325	0.014

Table 1.2.: Individual convex envelopes vs. extended formulation.

Using the work of Sherali and Adams [SA90, SA94, AS05], we derive extended formulations for the convex envelope of functions $f : [l^x, u^x] \times [l^y, u^y] \subseteq \mathbf{R}^{n_x} \times \mathbf{R}^{n_y} \to \mathbf{R}$, $(x, y) \mapsto f(x, y)$, where f is component-wise concave in the x-variables and further

- Class A: $n_y = 1$ and for all vertices v of $[l^x, u^x]$ it holds that $f(v, y)$ either is convex or concave in y over $[l^y, u^y]$.
- Class B: For all vertices v of $[l^x, u^x]$ it holds that $f(v, y)$ is convex in y over $[l^y, u^y]$.

Note that Class A contains the case of f being component-wise concave (*edge-concave*) in all variables for which the convex envelope is only known up to dimension three [MF05]. Moreover, Classes A and B contain special cases for which the convex envelope was recently derived by Khajavirad and Sahinidis [KS12a, KS12b]. We relate our work to their findings and discuss the advantages and disadvantages of the different approaches. Furthermore, we remark that the considered classes of functions are not only interesting from an academic point of view but also from a practical point of view. According to [KS12a] the two classes of functions account for at least 30 % of all the nonlinearities in the problem libraries GLOBALLib [GLO] and MINLPLib [BDM03] which contain many applications from engineering and science. Computational evidence of the proposed relaxations is given by the results of an ad-hoc implementation for component-wise concave functions and of a separator which we implemented for SCIP.

In Chapter 5 we explicitly study the simultaneous convexification of functions. In detail, we analyze the simultaneous convex hull of the graph of a vector of functions $f = (f_1, \dots, f_m) : \mathbf{R}^n \to \mathbf{R}^m$ over a *continuous* domain $D \subseteq \mathbf{R}^n$:

$$Q_D[f] := \operatorname{conv}\{(x, z) \in \mathbf{R}^{n+m} \mid (x, z) = (x, f(x)),\ x \in D\}.$$

We show that this concept has the potential to significantly improve the convexification even of univariate convex functions.

Example 1.3. Consider $f_1(x) = x^2$ and $f_2(x) = x^3$ over the domain $[l, u] = [1, 2]$. The individual convexifications of f_1 and f_2 lead to a relaxation with a volume of 0.1500 while the simultaneous convexification due to [KS53] yields a volume of only 0.0055. Thus, the gap between the two objects is 2627 %. ◇

General investigations of the simultaneous convex hull $Q_D[f]$ over continuous domains just started recently by Tawarmalani [Taw10] who analyzes the extreme points of $Q_D[f]$. In contrast to Tawarmalani's work, we establish a link between the simultaneous convex hull $Q_D[f] \subseteq \mathbf{R}^{n+m}$ and the individual convex hulls $Q_D[\sum_{i=1}^m \alpha_i f_i] = Q_D[\alpha^\intercal f] \subseteq \mathbf{R}^{n+1}$ with $\alpha \in \mathbf{R}^m$ via the relation

$$\begin{aligned} Q_D[f] &= \bigcap_{\alpha \in \mathbf{R}^m} \{(x, z) \in \mathbf{R}^{n+m} \mid (x, \alpha^\intercal z) \in Q_D[(\alpha^\intercal f)]\} \\ &= \bigcap_{\alpha \in \mathbf{R}^m} \{(x, z) \in \mathbf{R}^{n+m} \mid \operatorname{vex}_D[\alpha^\intercal f](x) \le \alpha^\intercal z,\ x \in D\}. \end{aligned} \tag{1.1}$$

This representation implies that the high dimensional object $Q_D[f] \subseteq \mathbf{R}^{n+m}$ can be described by the lower dimensional objects $Q_D[\alpha^\intercal f] \subseteq \mathbf{R}^{n+1}$ and, in particular, by the convex envelopes $\operatorname{vex}_D[\alpha^\intercal f](x)$. In other words this allows to exploit the knowledge of the well-studied concept of convex envelopes in order to derive $Q_D[f]$.

In this framework, we apply Equation (1.1) to characterize the extreme points of $Q_D[f]$ (inner description) and to determine necessary and sufficient subsets of $\alpha \in \mathbf{R}^m$ to describe $Q_D[f]$ via the constraints $\operatorname{vex}_D[\alpha^\intercal f](x) \le \alpha^\intercal z$ (outer description). In particular, we show that the union of the extreme points of $Q_D[\alpha^\intercal f]$ over all $\alpha \in \mathbf{R}^m$ is dense in the set of extreme points of $Q_D[f]$ w.r.t. the x-components. As $Q_D[\alpha^\intercal f]$ can be described by the convex envelopes $\operatorname{vex}_D[\alpha^\intercal f]$ and $\operatorname{vex}_D[-\alpha^\intercal f]$, we can thus take advantage of the existing theory of their "extreme points" to

describe the extreme points of $Q_D[f]$.

Regarding the $\alpha \in \mathbf{R}^m$ needed for $Q_D[f]$ in Equation (1.1), we identify two classes of cones whose interior points are not needed in the description of $Q_D[f]$. These cones are then explicitly derived for vectors of two and three univariate convex functions. Although the consideration of such "simple" functions may seem rather restrictive, strong relaxations for these vectors can have a significant impact on computations. For instance, higher dimensional functions, whose convex envelopes are not known, are often reformulated as sums and products of univariate (convex) functions as illustrated in Example 1.1. In such a setting the use of a simultaneous relaxation is clearly advantageous as indicated in Example 1.3.

Based on our analysis of the necessary $\alpha \in \mathbf{R}^m$ in the representation of $Q_D[f]$, we propose only a few $\alpha \in \mathbf{R}^m$ whose corresponding constraints $\mathrm{vex}_D[\alpha^\intercal f](x) \leq \alpha^\intercal z$ constitute a strong basic relaxation of $Q_D[f]$. For the vector of two univariate convex functions we can further provide a separation result which identifies for any $(\bar{x}, \bar{z}) \notin Q_D[f]$ an $\alpha \in \mathbf{R}^m$ such that the corresponding constraint $\mathrm{vex}_D[\alpha^\intercal f](x) \leq \alpha^\intercal z$ cuts off $(\bar{x}, \bar{z})$. To demonstrate the computational impact, we show the results of an ad-hoc implementation applied to an instance from GLOBALLib [GLO]. Motivated by the excellent computational results, we also implemented a separator in SCIP which is based on our proposed relaxations. This implementation clearly outperforms state-of-the-art global optimization solvers applied to a test set of 800 randomly generated instances.

CHAPTER 2

Bound Tightening for Material Balance Equations

In the process of modeling applications from chemical engineering, practitioners impose rather weak bounds on the variables in order to capture a large range of operating points. However, tighter bounds are often given implicitly by the constraint set. Thus, convex relaxations constructed over the original domains are unnecessarily weak and lead to long running times for global optimization solvers. To avoid this, most solvers apply *bound tightening* techniques to reduce the initial domains, e.g., BARON [TS05] and SCIP [Ach07].

This chapter introduces a bound tightening technique for a system of *material balance equations* which naturally occur in process modeling of multi-stage counter-current separation processes, as e.g., distillation, (melt-)crystallization, flotation, extraction, and membrane separations [CPW00]. The goal of such separation processes is to separate a given mixture into its components by means of a counter movement of two phases which posses different chemical and/or physical properties. Due to the different characteristics of the components they move with one of the phases, so that the separation takes place. Material balance equations ensure material conversation, that is, they require the material of one component to be the same for the different stages of the operational unit.

Material balance equations can be formally modeled as follows. Consider an operational unit consisting of N stages and two phases which are

called X and Y. Then, the material balance equations in the inner stages of the operating unit are given by

$$L_Y\ y_{i,l+1} - L_X\ x_{i,l} = L_Y\ y_{i,l} - L_X\ x_{i,l-1}, \qquad l = 2, \ldots, N-1, \tag{2.1}$$

where L_X and L_Y denote the flow-rates of the corresponding phases, and $x_{i,l}$ and $y_{i,l}$ denote the composition or concentration of component i at stage l in phase X and Y, respectively. To establish a link between the concentrations in the two phases, equilibrium functions are used in which $y_{i,l}$ is a function of $x_l = (x_{1,l}, \ldots, x_{k,l})$, i.e., $y_{i,l} = y_{i,l}(x_l)$. The concept of material balance equations is illustrated in Figure 2.1. The box represents the i-th stage of a separation unit and indicates the phase transition of the components. The arrows indicate the resulting incoming and outgoing material flows.

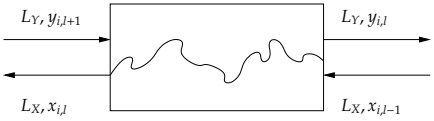


Figure 2.1.: Principle of material balance equations.

Computationally, optimization problems from chemical engineering, whose model formulations contain systems of material balance equations, are usually difficult to handle for global optimization solvers. Table 2.1 illustrates this for the two applications discussed in this thesis. The first application is a *hybrid distillation/melt-crystallization* process considered in this chapter, cf. instance T0 in Table 2.7. The second application is a *true moving bed* process introduced in Chapter 3, cf. Section 3.3.3. Both processes correspond to cost-intensive real world applications. For instance, in 2009 distillation columns consumed 6% of the overall U.S. energy production [Cah12]. Although the computations in Table 2.1 were accomplished by the state-of-the-art global optimization software BARON [TS05] (with default settings), the computational results are not satisfying and motivate further research in this area.

Two main reasons for the expensive computations can be identified. On the one hand, there is a large number of material balance equations resulting in a sparse model structure with respect to the occurrence of certain variables and functions. For instance, the composition variables

Application	Lower / upper bound	Gap
Distillation/Crystallization	12.69 / 154.00	168 %
True Moving Bed process	0.23 / 6.05	2530 %

Table 2.1.: Computational results after at least 100 hours.

$x_{i,l}$ and $y_{i,l}$ appear only in the material balance equations around stage l. The same is true for product terms like $L_X x_{i,l}$ and $L_Y y_{i,l}$, and the equilibrium functions $y_{i,l}(x_l)$. On the other hand, the domains for the concentration variables $x_{i,l}$ and $y_{i,l}$ are often rather weak.

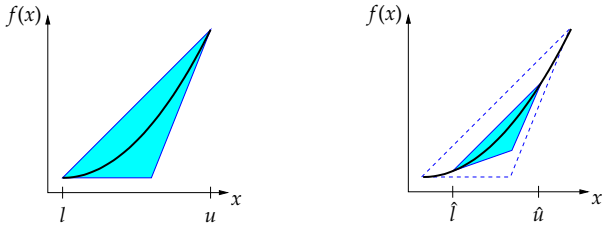
In this chapter we present a bound tightening technique for material balance equations in the framework of hybrid distillation/melt-crystallization processes that reduces the domains significantly and thus accelerates the computations. Our approach leads to *boundary intervals* for the composition variables $x_{i,l}$ of the distillation column which contain at least all feasible solutions of the original model. This forms the core of a small MINLP program used to relax the original problem. The small, relaxed MINLP formulation allows to check efficiently whether a given structure and domain can contain optimal solutions and thus reduces the domain of the original problem while guaranteeing that no globally optimal solution is lost. With the reduced domain at hand, the original problem is solved for global optimality. The computational results show that our approach considerably reduces the solution time. In particular, if the optimization of a stand-alone distillation column is considered, the solution time can be decreased by orders of magnitude.

This chapter is structured as follows. Initially, a review on existing bound tightening techniques is given in Section 2.1. In Section 2.2 we describe the basics of hybrid distillation/melt-crystallization processes and discuss a process model. In Section 2.3 we introduce the bound tightening technique for material balance equations and prove its computational impact in Section 2.4. This chapter is based on [BKK+] and [BKK+11].

2.1. Overview of Bound Tightening

The goal of bound tightening (BT) techniques is to reduce the domains of the variables. With this, tighter relaxations can be constructed over the resulting, smaller domains as indicated in Figure 2.2. This potentially

accelerates branch-and-bound algorithms, which is reflected by the various implementations of BT in several global optimization solvers, e.g., BARON [TS05], SCIP [Ach07], and COUENNE [BLL+09]. BT is also known as bound propagation, constraint propagation, domain filtering, domain reduction, and range reduction, cf. Section 1 in [BCLL12].



(a) Large domain. (b) Smaller domain.

Figure 2.2.: Impact of bound tightening on the relaxation quality. The blue shaded areas represent the convex relaxation of the graph of a function (bold black line).

In general, there are two classes of BT techniques [CL10]. Given an optimization problem $\min\{f_0(x) \mid x \in \mathcal{F} \cap [l,u]\}$, where $f_0 : \mathbf{R}^n \to \mathbf{R}$, $\mathcal{F} = \{x \mid f_i(x) \leq 0, i = 1,\ldots,m\} \subseteq \mathbf{R}^n$ is a closed convex set, and $[l,u] \subseteq \mathbf{R}^n$ is a box. *Feasibility Based Bound Tightening* (FBBT) aims at shrinking $[l,u]$ without excluding any feasible solution, i.e., determining the smallest box $[\hat{l},\hat{u}]$ with

$$\mathcal{F} \cap [l,u] = \mathcal{F} \cap [\hat{l},\hat{u}].$$

Optimality Based Bound Tightening (OBBT) shrinks $[l,u]$ without excluding any optimal solution. Let $f_0^\star$ be an upper bound (corresponding to the best known solution) on the problem. Then, the box $[\hat{l},\hat{u}]$ obtained by OBBT satisfies

$$\{x \in \mathcal{F} \mid f_0(x) \leq f_0^\star\} \cap [l,u] = \{x \in \mathcal{F} \mid f_0(x) \leq f_0^\star\} \cap [\hat{l},\hat{u}],$$

and excludes all solutions with an objective function value larger than $f_0^\star$. While FBBT uses the constraint set $\mathcal{F}$ to tighten the bounds, most OBBT methods rely on dual information and often solve auxiliary subproblems

to draw inferences on the domains, cf. [Sah03, BCLL12].

Feasibility Based Bound Tightening

The tightest bounds $[\hat{l}, \hat{u}]$ containing all feasible solutions can be computed by

$$\hat{l}_i = \min\{x_i \mid x \in \mathcal{F} \cap [l,u]\} \quad \text{and} \quad \hat{u}_i = \max\{x_i \mid x \in \mathcal{F} \cap [l,u]\}, \tag{2.2}$$

for all $i = 1, \ldots, n$. However, these auxiliary problems may be as hard as the original problem, cf. [BCLL12], so that this technique is rarely used in global optimization. In general, there are two main approaches to reduce the problem complexity and yet get improved bounds: (i) Relaxations of the optimization problems in (2.2) and (ii) BT by interval arithmetic.

The idea of the relaxation approach is to relax the set $\mathcal{F}$ by an easier description F such that the resulting optimization problems $\min / \max\{x_i \mid x \in F \cap [l,u]\}$ are efficiently to solve. One possibility is to use an arbitrary linear relaxation F which is discussed in [Kea06, LMR05] and applied in the software BARON. This approach works with the complete model and thus captures the overall problem characteristics. Yet, it requires to solve auxiliary optimization problems and relies on good linear relaxations.

BT by interval arithmetic exploits the dependencies among the variables imposed by each single constraint, cf. [Mes04]. Each constraint is decomposed into its basic algebraic operations and the variables are replaced by their intervals. The basic operations of interval arithmetic [Moo66] are defined as follows:

$$\begin{aligned}
[a,b] + [c,d] &= [a+c, b+d],\\
[a,b] - [c,d] &= [a-d, b-c],\\
[a,b] \cdot [c,d] &= [\min\{a\cdot c, a\cdot d, b\cdot c, b\cdot d\}, \max\{a\cdot c, a\cdot d, b\cdot c, b\cdot d\}],\\
[a,b]/[c,d] &= [a,b] \cdot [1/d, 1/c], \quad \text{if } 0 \notin [c,d].
\end{aligned}$$

While this approach might neglect some dependencies between the constraints, it needs only the cheap evaluation of interval arithmetic. Consider, for instance, a set $\mathcal{F} = \{x \in \mathbf{R}^2 \mid x_1 + x_2 = 2\}$ and $[l,u] = [0,2] \times [1,3]$. We want to check if this system implies tighter bounds on x_1. For this, we solve the constraint for x_1, i.e., $x_1 = 2 - x_2$, substitute x_2 by its given interval $[1,3]$, and intersect the resulting interval of x_1 with

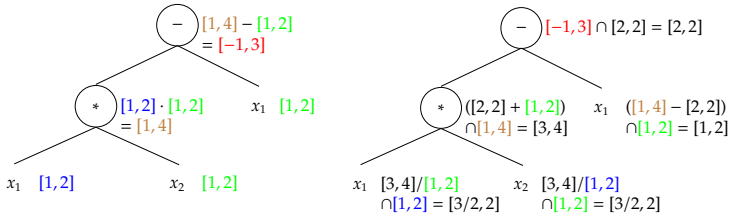
its given interval $[0, 2]$ which leads to

$$[\hat{l}_1, \hat{u}_1] = ([2, 2] - [1, 3]) \cap [0, 2] = [-1, 1] \cap [0, 2] = [0, 1].$$

Similarly, we obtain $[\hat{l}_2, \hat{u}_2] = [1, 2]$. This approach is well-known in the linear programming community and sometimes referred to as "poor man's linear programs". See [Sah03, Mes04, BCLL12] for details. An analogous procedure can be applied for nonlinear constraints, i.e., a given nonlinear constraint is solved for a variable and it is checked if better bounds are implied by means of interval arithmetic. See [Mes04] for a detailed discussion.

In optimization software BT by interval arithmetic is usually handled by *expression trees* [BGGP99, Mes04, BCLL12, Vig12] which is illustrated in the next example. In Section 2.3.1 we compare this standard approach to our developed BT technique.

Example 2.1. Let $\mathcal{F} = \{(x_1, x_2) \in \mathbf{R}^2 \mid x_1 x_2 - x_1 = 2\}$ and $[l, u] = [1, 2]^2$. The expression tree for the constraint is given in Figure 2.3. Each mathematical operation is represented by a node and the leaves of the tree correspond to the variables. In the *forward* propagation step the bounds on the variables



(a) Forward propagation (up).

(b) Backward propagation (down).

Figure 2.3.: Expression tree evaluation for the constraint $x_1 x_2 - x_1 = 2$ with $(x_1, x_2) \in [1, 2]^2$.

are propagated upwards through the tree by means of interval arithmetic to obtain bounds on the constraints, cf. Figure 2.3 (a). In the *backward* propagation the computed bounds on the constraints are intersected with the original bounds at the root node of the expression tree. This bound is

then propagated downwards to the leaves of the expression tree using the *inverse* arithmetic operations, cf. Figure 2.3 (b). For instance, the bounds on the expression x_1x_2 are computed to be $[3,4]$ and $(x_1,x_2) \in [1,2]^2$. By interval arithmetic we obtain tighter bounds on x_1, x_2 via $[3,4]/[1,2] = [3,4] \cdot [1/2,1] = [3/2,4]$ which needs to be intersected with the original intervals $[1,2]$ leading to the domains $[3/2,2]$ for both x_1 and x_2. ◇

The example shows how BT by interval arithmetic can be easily automatized by expression trees. Nevertheless, it cannot be guaranteed that the tightest possible bounds are obtained by this procedure. One reason is the representation of the constraints and the conservativeness of the approach. For example, the expression $x - x$ with $x \in [0,1]$ is replaced by $[0,1] - [0,1] = [-1,1]$ while $x - x = 0$. To avoid these problems, some authors recently started to utilize the particular structure of a constraint, e.g., quadratic constraints [DN10, BHV09, Vig12]. Another reason is the individual and consecutive treatment of the constraints so that not all interactions within the constraint set are exploited. As BT by interval arithmetic is computationally cheap, the procedure is often applied in a loop. Yet, this procedure does not necessarily converge to the tightest box, see e.g., [BCLL12].

Optimality Based Bound Tightening

In optimization one is often not interested in all feasible solutions but in an optimal solution. Therefore, local information regarding the objective function value can be used to cut-off regions with a worse objective function value. For instance, consider the problem $\min\{x_1 + x_2 \mid x_1^2x_2 + x_2 \geq 3, (x_1,x_2) \in [0,3]^2\}$ with the feasible solution $(x_1,x_2) = (2,0.6)$ corresponding to the objective function value of 2.6. One could add the constraint $x_1 + x_2 \leq 2.6$ to the constraint set and apply FBBT which in this case tightens the upper bounds on the two variables to 2.6.

Ryoo and Sahinidis [RS95, RS96] developed an OBBT technique which is not based on FBBT but on information from the dual solution of a convex relaxation.

Corollary 2.2 (cf. Corollary 2 in [RS95])**.** *Given a minimization problem P. Let R be a convex programming relaxation of P with an optimal objective function value of L and consider a range constraint $x_j \leq u_j$ that is active at the solution of Problem R with a dual multiplier value $\lambda_j > 0$. Let U be a known*

upper bound for problem P. Then, the following constraint is valid for P.

$$x_j \geq \max\{l_j, u_j - (U - L)/\lambda_j\}.$$

An analogous result can be used to tighten the upper bound on variables x_j.

Corollary 2.2 is only applicable to variables whose solution in the relaxed program is attained at their lower or upper bound. But this idea can be extended to other variables by *probing* (cf. [RS95]). Here, a variable is fixed to one of its bounds and then the convex relaxation is solved and Corollary 2.2 can be applied.

Remark 2.3. Some authors use the terms FBBT and OBBT but refer to different concepts, e.g., [BLL+09, BCLL12]. They distinguish between techniques which require the solution of auxiliary optimization problems (OBBT) and techniques which use only the available information about the constraint set or current solutions (FBBT).

2.2. Hybrid Distillation/Melt-Crystallization Processes

Separation of closely boiling mixtures is a challenging problem in process synthesis and design. A typical example is the separation of mixtures of isomers like n/iso-aldehyde mixtures arising from oxo-synthesis. Standard distillation is often not favorable due to high process costs, in particular for long-chain molecules [MBR+11]. A more energy and cost efficient separation process for such closely boiling mixtures is thus desirable and may be obtained by an optimal combination of *distillation* and *melt-crystallization* taking advantage of both processes (see [FNN+08]).

Distillation is a separation process which exploits the different boiling temperatures of the single components of a mixture. *Distillation columns* are used to perform this separation. The liquid mixture is fed into the distillation column, where it is heated up. The components with the lower boiling temperature evaporate and move upwards with the vapor phase while the components with the higher boiling temperatures remain in the liquid phase and are withdrawn at the bottom of the distillation column.

Melt-crystallization is based on the different composition properties of the components in their liquid and solid (crystal) state. When cooling a mixture down to a certain temperature, some components start to form

relatively pure crystals. The crystal grow depends on the composition of the components in the mixture and requires a continuous supersaturation of the specific components. As crystal structures are highly complex, the atoms of the other components do often not fit into the crystal structure of the supersaturated components and stay in the melt. After the growth of the crystals the melt is withdrawn from the operation unit and only the pure crystals remain, leading to the separation of the components. See [CPW00] for further details on both techniques.

The optimal design of hybrid distillation/melt-crystallization processes with structural and operational degrees of freedom can be modeled as a mixed-integer nonlinear program, which is usually difficult to solve due to nonconvex nonlinearities and integrality conditions on some variables. Over the last years several methods for the local optimization of standalone distillation column models have been established and extended to more complex superstructures (e.g., see [VG90, VG93, YG00, BA02, BA03, KKM09]). For the preprocessing these methods usually make use of shortcut procedures to rank different design alternatives, provide good initial solutions, and give improved bounds on key variables. For a comprehensive list of different shortcut evaluation methods for distillation columns, we refer to the work [LAT08] and the references therein.

Several studies show that rigorous optimization with a good preprocessing, e.g., good initial solutions and a reduced search space, can determine locally optimal solutions efficiently. However, Grossmann et al. [GAB05] conclude that global optimization of such processes is still a major challenge (see also [GKH+06, GHJ+08a, BGSA08]).

An example for the optimization of hybrid distillation/melt-crystallization processes for separating ternary mixtures is given in Franke et al. [FNN+08]. Therein, the authors first apply heuristics to determine different process configurations that are then evaluated by shortcut methods with respect to their energy requirements. Finally, only the most promising process configurations are rigorously optimized with respect to an economic objective function. The authors apply a modified Generalized Bender's Decomposition algorithm (cf. [Flo95]) which cannot generally prove global optimality.

The discussion above shows that the global optimization of hybrid distillation/melt-crystallization processes as well as their individual operations is still an open issue in the process engineering community. The aim of the remaining section is to introduce a suitable model description which is investigated in Section 2.3.

Process Model

Subsequently, we describe a simple model for a binary separation combining distillation and melt-crystallization. This is a first step to develop methods suitable for more realistic hybrid separation process models with more detailed objective functions.

Distillation Column

As a starting point we use a MINLP model formulation for a continuous, counter-current distillation in the line of Viswanathan and Grossmann [VG93] and we refer to it as *reference* model formulation. The model assumes steady-state, simple thermodynamics with constant relative volatilities, a total condenser and reboiler, a single saturated liquid feed flow and constant molar overflows. The basic concepts of this model are introduced with the help of Figure 2.4. The liquid mixture is fed into

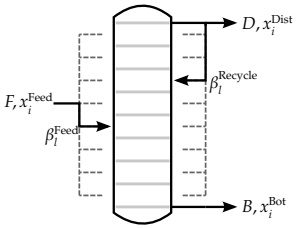


Figure 2.4.: Model structure with variable positions of feed and condenser recycle flow.

the column with a certain flow F and a composition x_i^{F} for each component $i \in \{A, B\}$. The mixture is then heated up and the component with the lower boiling point evaporates and moves upwards in the vapor phase with flow V and composition $y_{i,l}$, where the index l denotes the tray of the column. The trays are labeled downwards, i.e., the top (condenser) is labeled $l = 1$ and the bottom (reboiler) is labeled $l = N_{\text{trays}}^{\max}$. The component with the higher boiling point remains in the liquid phase, whose flow and composition are denoted by L_l and $x_{i,l}$, respectively. The enriched product streams are withdrawn at the top and bottom of the column with a flow and a composition of D and x_i^{Dist}, and B and x_i^{Bot}, respectively. The

binary variables β_l^{F} and $\beta_l^{\mathrm{Recycle}}$ indicate whether the feed or the reflux is introduced at tray l or not. The latter condition determines the number of active trays, i.e., the column length. The model of the distillation column is given as follows.

Component material balance with the reflux flow R:

$$(y_{i,l+1} - y_{i,l})\, V + x_{i,l-1} L_l - x_{i,l} L_{l+1} + x_i^{\mathrm{F}}\, F\, \beta_l^{\mathrm{F}} + x_i^{\mathrm{Dist}}\, R\, \beta_l^{\mathrm{Recycle}} = 0. \tag{2.3}$$

Total material balance:

$$L_{l+1} = L_l + F\beta_l^{\mathrm{F}} + R\beta_l^{\mathrm{Recycle}} \quad \text{with} \quad L_1 = 0. \tag{2.4}$$

Condenser and column total material balance:

$$0 = V - D - R \quad \text{and} \quad 0 = F - D - B. \tag{2.5}$$

Vapor-liquid equilibrium with given relative volatilities α_i:

$$y_{i,l} = \frac{\alpha_i x_{i,l}}{\sum_{j=1}^{N_{\mathrm{comp}}} \alpha_j x_{j,l}}. \tag{2.6}$$

Single recycle and feed location, and feed below recycle location:

$$\sum_{l=1}^{N_{\mathrm{trays}}^{\max}} \beta_l^{\mathrm{Recycle}} = 1, \quad \sum_{l=1}^{N_{\mathrm{trays}}^{\max}} \beta_l^{\mathrm{F}} = 1, \quad \sum_{l=1}^{N_{\mathrm{trays}}^{\max}} l\, \beta_l^{\mathrm{F}} \geq \sum_{l=1}^{N_{\mathrm{trays}}^{\max}} l\, \beta_l^{\mathrm{Recycle}}. \tag{2.7}$$

Total condenser and reboiler:

$$x_i^{\mathrm{Dist}} = y_{i,1}, \qquad x_i^{\mathrm{Bot}} = y_{i,N_{\mathrm{trays}}^{\max}+1} = x_{i,N_{\mathrm{trays}}^{\max}}. \tag{2.8}$$

Summation conditions:

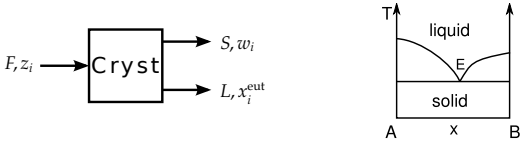
$$\sum_{i=1}^{N_{\mathrm{comp}}} x_{i,l} = 1, \quad \sum_{i=1}^{N_{\mathrm{comp}}} y_{i,l} = 1. \tag{2.9}$$

It follows from the definition of the binary variables that the column length, i.e., the number of active trays N_{active}, is given by

$$N_{\text{active}} := N_{\text{trays}}^{\max} + 1 - \sum_{l=1}^{N_{\text{trays}}^{\max}} l\,\beta_l^{\text{Recycle}}. \tag{2.10}$$

Crystallizer Model

The crystallizer model incorporates ideal behavior of a binary, eutectic system. The liquid mixture is fed into the crystallizer with flow F and composition z_i, $i \in \{A, B\}$. See Figure 2.5 (a). We assume an eutectic



(a) Principle of a crystallizer.

(b) Phase diagram.

Figure 2.5.: (a) Input and output streams of a crystallizer. (b) A qualitative phase diagram for a binary, eutectic system. The point E is called eutectic point, and represents the eutectic temperature and eutectic composition x_i^{eut}.

system, i.e., the crystallization is done at the eutectic temperature, and depending on the relation of the input composition z_i to the given eutectic composition x_i^{eut} either component A or component B forms a pure crystal. See Figure 2.5 (b). Consequently, we have one outlet stream with pure crystals and composition $w_i \in \{0, 1\}$, and another outlet stream of remainder liquid with eutectic composition x_i^{eut}.

The crystallizer is then modeled by the following equation.

$$Fz_i = Sw_i + Lx_i^{\text{eut}}, \quad i = \text{A}, \text{B}, \qquad \text{where} \quad w_i = \begin{cases} 1, & \text{if } z_i \geq x_i^{\text{eut}}, \\ 0, & \text{if } z_i < x_i^{\text{eut}}. \end{cases} \tag{2.11}$$

Superstructure

We consider hybrid distillation/melt-crystallization processes consisting possibly of two crystallizers and one distillation column for a binary separation task as layouted in Figure 2.6. A set of binary variables determines the existence of connections between the process units (dashed lines). Additional constraints are added to enforce that output flows are transferred completely to a single target.

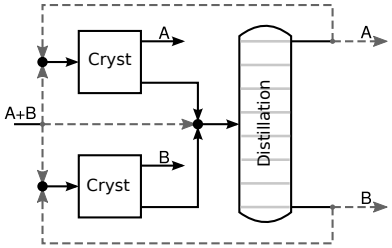
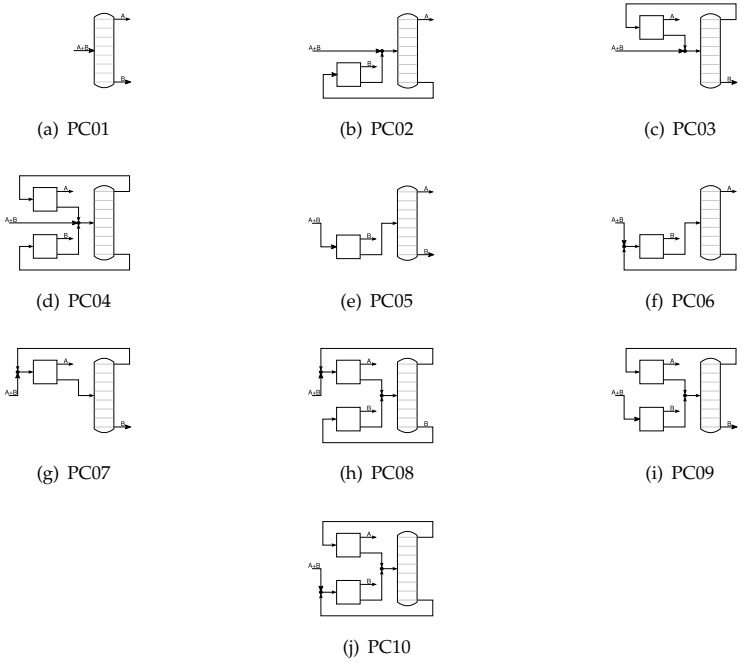


Figure 2.6.: Process structure consisting of a distillation column (vertical tube) and two crystallizers (squares). The dashed lines reflect possible material flows which can be enabled or disabled by binary variables. The black lines correspond to streams which are always enabled.

The presented superstructure contains ten meaningful process configurations, which are listed in Figure 2.7. For instance, Figure 2.7 (c) displays a process configuration of one crystallizer and the distillation column, where the mixture is fed into the distillation column. The product stream withdrawn at the reboiler (bottom) is enriched with component B such that it is satisfies the purity requirements. The product stream at the condenser (top) is recycled into the crystallizer, where pure crystals of component A are generated. The remaining liquid mixture is withdrawn from the crystallizer, mixed with the feed mixture, and fed into the distillation column in a continuous process.

Process Costs

A simplified cost function is used accounting for essential cost factors. The overall annual costs of the general process consist of annualized



(a) PC01 (b) PC02 (c) PC03

(d) PC04 (e) PC05 (f) PC06

(g) PC07 (h) PC08 (i) PC09

(j) PC10

Figure 2.7.: List of possible process configurations.

investment costs and operating costs. The investment costs k_1 of the column are proportional to the column length N_{active} and also to the vapor flow V since higher flows require a larger column diameter. The operating costs k_2 are proportional to the desired vapor flow V. For the crystallizer, investment and operating costs are proportional to the feed flow $F^{j_{\text{Cr}}}$ and both are covered by k_3 since the crystallizer size is fixed here. Thus, the objective function evolves to

$$\text{cost} \quad = \quad (k_1 N_{\text{active}} V + k_2 V) + k_3 \sum_{j_{\text{Cr}}=1}^{2} F^{j_{\text{Cr}}}. \tag{2.12}$$

Evaluation of the Process Model from an Optimization Point of View

The mathematical formulation of the process model involves two main types of nonlinearities: Product terms of two or three variables, that appear in the material balance equations (2.3) and the cost function (2.12), and fractions of linear terms in the vapor-liquid equilibrium in Equation (2.6). As we consider only binary mixtures the vapor-liquid equilibrium can be reduced to univariate convex or concave functions via the summation condition $x_{A,l} + x_{B,l} = 1$ in Equation (2.9), e.g., $y_{A,l} = (\alpha_A x_{A,l})/(\alpha_A x_{A,l} + \alpha_B(1 - x_{A,l}))$ is convex. For both types of nonlinearities explicit formulas for the best convex under- and concave overestimators are known (e.g., see [McC76, MF03]) and state-of-the-art global optimization software can handle each single nonlinearity of such type very well. However, a main factor influencing the relaxation quality of these estimators is given by the underlying domain of the involved variables. It is therefore desirable to work with the tightest possible bounds on each variable. The initial bounds on all composition variables $x_{i,l}$ and $y_{i,l}$ are given by zero and one even though tighter bounds are induced by the constraint set, as we show in the following section.

2.3. Global Optimization Techniques for Distillation Columns

For our new approach to globally optimize hybrid distillation/melt-crystallization processes we proceed as follows. In Section 2.3.1 we present a BT technique for the composition variables related to a binary distillation column. In contrast to general purpose BT techniques, we explicitly exploit the analytical properties of material balance equations. Based on the proposed technique, a relaxed MINLP model for the distillation column is defined in Section 2.3.2. We apply this model in our computations to identify infeasible or nonoptimal process configurations and/or operating subdomains. In Section 2.3.3 we use the relaxed MINLP model to derive an alternative model formulation for distillation columns, which allows us to assign the improved bounds from the proposed BT technique directly to the composition variables.

2.3.1. Bound Tightening

Distillation columns are applied to produce highly enriched product streams which are withdrawn at the top (condenser) and bottom (reboiler) of the column. Thus, the compositions of the product streams usually have to satisfy high purity requirements, this is, there are tight bounds on the corresponding composition variables. Given these bounds at the top and bottom of the distillation column, we use tray-to-tray calculations to propagate the bounds through the distillation column for a given range of specifications. A similar procedure can be applied to improve initial bounds on concentration variables for MINLPs arising from true moving bed processes and is discussed in Chapter 3.

We investigate *binary* distillation processes and assume that the initial lower and upper bounds on the composition variables $x_{i,l}$ and $y_{i,l}$ are the trivial bounds zero and one, respectively. Our point of departure is a distillation column with a fixed column length and a fixed position of the feed stage, i.e., the binary variables β_l^{Recycle} and β_l^{F} are fixed. These assumptions simplify the component material balance equations (2.3) to

$$\begin{aligned}
Dx_i^{\text{Dist}} &= Vy_{i,l+1} - Rx_{i,l}, && l = 1,\\
Vy_{i,l} - Rx_{i,l-1} &= Vy_{i,l+1} - Rx_{i,l} && l = 2,\dots,l^{\text{F}}-1,\\
Vy_{i,l} - Rx_{i,l-1} &= Vy_{i,l+1} - (R+F)x_{i,l} + Fx_i^{\text{F}}, && l = l^{\text{F}},\\
Vy_{i,l} - (R+F)x_{i,l-1} &= Vy_{i,l+1} - (R+F)x_{i,l}, && l = l^{\text{F}}+1,\dots,N_{\text{trays}}^{\max}-1,\\
Vy_{i,l} - (R+F)x_{i,l-1} &= -Bx_i^{\text{Bot}}, && l = N_{\text{trays}}^{\max},
\end{aligned} \tag{2.13}$$

which reveal the structure of the general material balance equations presented in Equation (2.1). Figure 2.8 illustrates the material flow described in Equation (2.13) and depicts also the three parts of a distillation column: The rectifying section ($l = 1,\dots,l^{\text{F}}-1$), the feed section ($l = l^{\text{F}}$), and the stripping section ($l = l^{\text{F}}+1,\dots,N_{\text{trays}}^{\max}$). Note that the enriched phases are withdrawn at the end of the rectifying and stripping section, i.e., the composition variables x_i^{Dist} and x_i^{Bot} have to satisfy the desired purity requirements so that their bounds are tight. The structure of the material balance equations allows us to propagate these tight bounds through the distillation column.

We now consider the rectifying and the stripping section separately, as if both sections constitute independent operational units. In Section 2.3.2 and 2.3.3 we show how the two units can be coupled again. We introduce

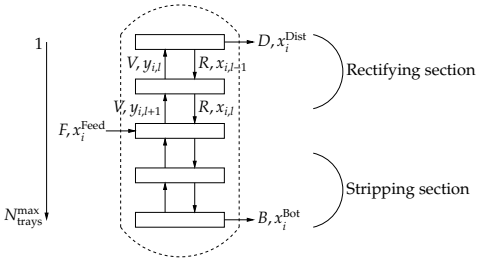


Figure 2.8.: Schematic representation of a distillation column with fixed feed position and fixed length.

the composition variables $x_{i,l}^{\text{rect}}$, $y_{i,l}^{\text{rect}}$ and $x_{i,l}^{\text{strip}}$, $y_{i,l}^{\text{strip}}$ for the rectifying and stripping section, respectively. The total material balance equations (2.5) state $R = V - D$ and $F = B + D$ so that $F + R = V + B$ and thus, the component material balance equations for the two sections can be derived from Equations (2.13)

$$\begin{aligned} D\, x_i^{\text{Dist}} &= V\, y_{i,l+1}^{\text{rect}} - (V - D)\, x_{i,l}^{\text{rect}}, \qquad l = 1, \ldots, l_{\max}^{\text{rect}} - 1, \\ B\, x_i^{\text{Bot}} &= (V + B)\, x_{i,l+1}^{\text{strip}} - V\, y_{i,l}^{\text{strip}}, \qquad l = 1, \ldots, l_{\max}^{\text{strip}} - 1, \end{aligned} \tag{2.14}$$

where $l_{\max}^{\text{rect}}$ and $l_{\max}^{\text{strip}}$ denote the length of the two column sections. The trays of the rectifying section are numbered from the condenser downwards while the trays of the stripping section are numbered upwards. Thus, given the bounds from the purity requirements on the product compositions x_i^{Dist} and x_i^{Bot} we obtain bounds on the composition variables $x_{i,1}^{\text{rect}}$,$y_{i,1}^{\text{rect}}$ and $x_{i,1}^{\text{strip}}$,$y_{i,1}^{\text{strip}}$ by the initial conditions corresponding to Equation (2.8), i.e., $x_i^{\text{Dist}} = y_{i,1}^{\text{rect}}$ and $x_i^{\text{Bot}} = x_{i,1}^{\text{strip}}$. These bounds are propagated through the column sections via the following tray-to-tray calculations which are a reformulation of Equation (2.14) (cf. Equations (1a) and (1b) in [LVD85]):

$$\begin{aligned} y_{i,l+1}^{\text{rect}} &= x_{i,l}^{\text{rect}} + \tfrac{D}{V}(x_i^{\text{Dist}} - x_{i,l}^{\text{rect}}) &&=: \Phi_{i,l+1}^{y}\left(x_{i,l}^{\text{rect}}, x_i^{\text{Dist}}, D, V\right), \\ x_{i,l+1}^{\text{strip}} &= \tfrac{1}{V+B}(V y_{i,l}^{\text{strip}} + B x_i^{\text{Bot}}) &&=: \Phi_{i,l+1}^{x}(y_{i,l}^{\text{strip}}, x_i^{\text{Bot}}, B, V). \end{aligned} \tag{2.15}$$

for $l = 1, \ldots, l^{\text{rect}}_{\max} - 1$ and $l = 1, \ldots, l^{\text{strip}}_{\max} - 1$.

Subsequently, we present the bound propagation for component A which is enriched in the vapor phase and withdrawn at the top of the distillation column. The bounds for component B can be derived in an analogous procedure. As $\Phi^{y}_{\text{A},l+1}$ is monotonous in its variables, we can use the bounds on the variables x^{Dist}_i, D, V to compute lower bounds "lb" and upper bounds "ub" on $y^{\text{rect}}_{\text{A},l+1}$. To obtain bounds on $x^{\text{rect}}_{\text{A},l+1}$, we exploit the summation condition for binary mixtures $x^{\text{rect}}_{\text{A},l+1} + x^{\text{rect}}_{\text{B},l+1} = 1$ and write the vapor-liquid equilibrium $y^{\text{rect}}_{\text{A},l+1}$ as a univariate and monotonously increasing function in $x^{\text{rect}}_{\text{A},l+1}$, namely $y^{\text{rect}}_{\text{A},l+1} = \alpha_{\text{A}} x^{\text{rect}}_{\text{A},l+1} / ((\alpha_{\text{A}} - \alpha_{\text{B}}) x^{\text{rect}}_{\text{A},l+1} + \alpha_{\text{B}})$.

Proposition 2.4 (BT for the rectifying section)**.** *Given nonnegative domains for the variables x^{Dist}_i, D, a positive domain for V, and* $\text{ub}(D) \le \text{lb}(V)$*. Then, for* $l = 1, \ldots, l^{\text{rect}}_{\max} - 1$

$$\begin{aligned} \text{lb}(y^{\text{rect}}_{\text{A},l+1}) &= \min\left\{0\ ,\ \Phi^{y}_{\text{A},l+1}\left(\text{lb}(x^{\text{rect}}_{\text{A},l}), \text{lb}(x^{\text{Dist}}_{\text{A}}), \text{lb}(D), \text{ub}(V)\right)\right\}, \\ \text{ub}(y^{\text{rect}}_{\text{A},l+1}) &= \max\left\{1\ ,\ \Phi^{y}_{\text{A},l+1}\left(\text{ub}(x^{\text{rect}}_{\text{A},l}), \text{ub}(x^{\text{Dist}}_{\text{A}}), \text{ub}(D), \text{lb}(V)\right)\right\}. \end{aligned} \tag{2.16}$$

If $y^{\text{rect}}_{\text{A},l} = \alpha_{\text{A}} x^{\text{rect}}_{\text{A},l} / ((\alpha_{\text{A}} - \alpha_{\text{B}}) x^{\text{rect}}_{\text{A},l} + \alpha_{\text{B}})$*, then for* $l = 1, \ldots, l^{\text{rect}}_{\max} - 1$

$$\begin{aligned} \text{lb}(x^{\text{rect}}_{\text{A},l+1}) &= \frac{\alpha_B \,\text{lb}(y^{\text{rect}}_{\text{A},l+1})}{\alpha_A - (\alpha_A - \alpha_B)\,\text{lb}(y^{\text{rect}}_{\text{A},l+1})}, \\ \text{ub}(x^{\text{rect}}_{\text{A},l+1}) &= \frac{\alpha_B \,\text{ub}(y^{\text{rect}}_{\text{A},l+1})}{\alpha_A - (\alpha_A - \alpha_B)\,\text{ub}(y^{\text{rect}}_{\text{A},l+1})}. \end{aligned} \tag{2.17}$$

Proof. The first partial derivatives of $\Phi^{y}_{A,l+1}\left(x^{\text{rect}}_{A,l}, x^{\text{Dist}}_{A}, D, V\right)$ read

$$\frac{\partial \Phi^{y}_{A,l+1}}{\partial x^{\text{rect}}_{A,l}} = \frac{V-D}{V}, \quad \frac{\partial \Phi^{y}_{A,l+1}}{\partial x^{\text{Dist}}_{A}} = \frac{D}{V}, \quad \frac{\partial \Phi^{y}_{A,l+1}}{\partial D} = \frac{x^{\text{Dist}}_{A} - x^{\text{rect}}_{A,l}}{V}, \quad \frac{\partial \Phi^{y}_{A,l+1}}{\partial V} = -\frac{D(x^{\text{Dist}}_{A} - x^{\text{rect}}_{A,l})}{V^2}.$$

Note that (i) $V - D \ge 0$ if $\text{lb}(V) \ge \text{ub}(D)$, (ii) $D \ge 0, V > 0$, and (iii) $(x^{\text{Dist}}_{A} - x^{\text{rect}}_{A,l}) \ge 0$. The first two conditions hold in general while the third condition holds only for component A. It reflects the monotonicity in the extreme components, i.e., the composition of component A decreases while moving from top to bottom of the column which guarantees that $(x^{\text{Dist}}_{A} - x^{\text{rect}}_{A,l}) \ge 0$ for all $l = 1, \ldots, l^{\text{rect}}_{\max}$. See [FK]. Then, the first three partial

derivatives are nonnegative while the fourth is nonpositive. This implies monotonicity in all variables and leads to Equation (2.16).

The equilibrium function $y^{\text{rect}}_{\text{A},l} = \alpha_{\text{A}} x^{\text{rect}}_{\text{A},l} / ((\alpha_{\text{A}} - \alpha_{\text{B}}) x^{\text{rect}}_{\text{A},l} + \alpha_{\text{B}})$ is monotonously increasing in $x^{\text{rect}}_{\text{A},l}$ because $\frac{\partial y^{\text{rect}}_{\text{A},l}}{\partial x^{\text{rect}}_{\text{A},l}} = \frac{\alpha_A \alpha_B}{((\alpha_A - \alpha_B) x^{\text{rect}}_{\text{A},l} + \alpha_B)^2} \geq 0$ as $\alpha_A, \alpha_B \geq 0$. Hence, $\text{lb}(y^{\text{rect}}_{\text{A},l}) = y^{\text{rect}}_{\text{A},l}(\text{lb}(x^{\text{rect}}_{\text{A},l}))$ and $\text{ub}(y^{\text{rect}}_{\text{A},l}) = y^{\text{rect}}_{\text{A},l}(\text{ub}(x^{\text{rect}}_{\text{A},l}))$. Using the inverse function $(y^{\text{rect}}_{\text{A},l})^{-1} = x^{\text{rect}}_{\text{A},l} = (\alpha_B y^{\text{rect}}_{\text{A},l}) / ((\alpha_A - (\alpha_A - \alpha_B) y^{\text{rect}}_{\text{A},l})$, Equation (2.17) is implied. □

The BT technique for the stripping section is based on the same ideas but does not rely on the explicit description of the vapor-liquid equilibrium functions but rather on their monotonicity properties.

Proposition 2.5 (BT for the stripping section)**.** *Given nonnegative domains for the variables x_i^{Bot}, B and a positive domain for V. Then, for $l = 1, \ldots, l^{\text{strip}}_{\max} - 1$*

$$\begin{aligned} \text{lb}(x^{\text{strip}}_{\text{A},l+1}) &= \min\left\{0\ ,\ \Phi^x_{\text{A},l+1}\left(\text{lb}(y^{\text{strip}}_{\text{A},l}), \text{lb}(x^{\text{Bot}}_{\text{A}}), \text{ub}(B), \text{lb}(V)\right)\right\}, \\ \text{ub}(x^{\text{strip}}_{\text{A},l+1}) &= \max\left\{1\ ,\ \Phi^x_{\text{A},l+1}\left(\text{ub}(y^{\text{strip}}_{\text{A},l}), \text{ub}(x^{\text{Bot}}_{\text{A}}), \text{lb}(B), \text{ub}(V)\right)\right\}. \end{aligned} \tag{2.18}$$

If $y^{\text{strip}}_{\text{A},l+1}$ is monotonously increasing in $x^{\text{strip}}_{\text{A},l+1}$ and monotonously decreasing in $x^{\text{strip}}_{\text{B},l+1}$, then for $l = 1, \ldots, l^{\text{strip}}_{\max} - 1$

$$\begin{aligned} \text{lb}(y^{\text{strip}}_{\text{A},l+1}) &= y^{\text{strip}}_{\text{A},l+1}\left(\text{lb}(x^{\text{strip}}_{\text{A},l+1}), 1 - \text{lb}(x^{\text{strip}}_{\text{A},l+1})\right), \\ \text{ub}(y^{\text{strip}}_{\text{A},l+1}) &= y^{\text{strip}}_{\text{A},l+1}\left(\text{ub}(x^{\text{strip}}_{\text{A},l+1}), 1 - \text{ub}(x^{\text{strip}}_{\text{A},l+1})\right). \end{aligned} \tag{2.19}$$

Proof. The first partial derivatives of $\Phi^x_{A,l+1}(y^{\text{strip}}_{A,l}, x^{\text{Bot}}_A, B, V)$ read

$$\frac{\partial \Phi^x_{A,l+1}}{\partial y^{\text{strip}}_{A,l}} = \frac{V}{V+B},\ \frac{\partial \Phi^x_{A,l+1}}{\partial x^{\text{Bot}}_A} = \frac{B}{V+B},\ \frac{\partial \Phi^x_{A,l+1}}{\partial B} = \frac{V(x^{\text{Bot}}_A - y^{\text{strip}}_{A,l})}{(V+B)^2},\ \frac{\partial \Phi^x_{A,l+1}}{\partial V} = -\frac{B(x^{\text{Bot}}_A - y^{\text{strip}}_{A,l})}{(V+B)^2}.$$

Similar to the proof of Proposition 2.4 it holds that (i) $B \geq 0, V > 0$ and (ii) $(x^{\text{Bot}}_A - y^{\text{strip}}_{A,l}) \leq 0$. Therefore, $\Phi^x_{\text{A},l-1}$ is monotonously increasing in $y^{\text{strip}}_{\text{A},l}, x^{\text{Bot}}_i$, and V, and monotonously decreasing in B which implies Equation (2.18).

For the bounds on $y^{\text{strip}}_{\text{A},l+1}$ we exploit monotonicity of $y^{\text{strip}}_{\text{A},l+1}$ and the summation condition $1 = x^{\text{strip}}_{\text{A},l+1} + y^{\text{strip}}_{\text{B},l+1}$. For clarity, we skip some indices. Then,

$y_A(x_A, x_B) = y_A(x_A, 1 - x_A)$, whose first partial derivative with respect to x_A is nonnegative. □

To summarize, Formulas (2.17) and (2.18) represent the boundary intervals for the composition variables in the two column sections for given lower and upper bounds on the variables $x_i^{Dist}, x_i^{Bot}, D, B, V$. In contrast to general BT techniques, we take advantage of the analytical properties of the underlying constraints, namely the mass balance equations, to derive tighter bounds on some variables. Moreover, we make use of the knowledge of the distillation processes, i.e., we start the bound propagation at the outlet ports and move from one tray to the next whereas general BT technique choose the constraints and variables rather randomly.

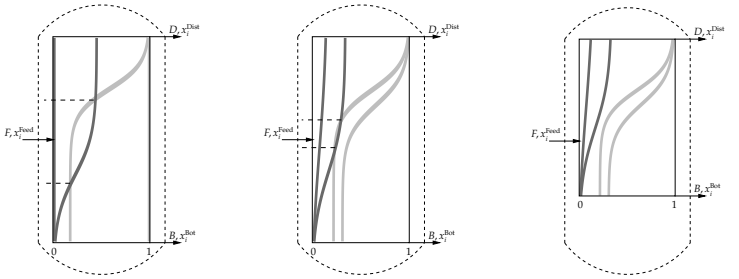
Illustration For the purpose of illustration consider a distillation column with specifications as given in Table 2.2. From the relations $F = D + B$ and $Fx_i^F = Dx_i^{Dist} + Bx_i^{Bot}$, we can derive strong bounds on D, B, $x_{A,1}^{rect}$,$y_{A,1}^{rect}$, $x_{A,1}^{strip}$ and $y_{A,1}^{strip}$ (see Table 2.2).

Given specifications		Implied bounds	
F	1	D	[0.24,0.25]
(x_A^F, x_B^F)	(0.25,0.75)	B	[0.74,0.75]
(α_A, α_B)	(1.3,1)	$x_{A,1}^{rect}$	[0.98,1.00]
$pur_A : x_A^{Dist} = y_{A,1}^{rect}$	≥ 0.99	$y_{A,1}^{rect}$	[0.99,1.00]
$pur_B : x_B^{Bot} = x_{B,1}^{strip}$	≥ 0.99	$x_{A,1}^{strip}$	[0.00,0.01]
$(l_{max}^{rect}, l_{max}^{strip})$	(75,75)	$y_{A,1}^{strip}$	[0.00,0.02]
V	[2,5]		

Table 2.2.: Specifications for a distillation column and implied bounds.

Figure 2.9 (a) displays the impact of the BT technique for the composition variables associated with component A assuming the setting of Table 2.2. The original bounds on all $x_{A,l}$ are given by zero and one. These bounds are indicated in Figure 2.9 (a) by the rectangle within the distillation column. The improved bounds on the $x_{A,l}$-variables obtained by applying the recursive formulas are given by the light gray lines and the dark gray lines. The two dashed lines at the intersection of the boundary

intervals represent the range for a possible feed position if 75 trays are active.



(a) Parameters and domains according to Table 2.2.

(b) Stronger bounds: $3 \leq V \leq 4$, $x_{\text{A},1} \leq 0.99$, $x_{\text{A},N} \geq 0.005$

(c) Shorter column: The BT technique yields a conflict at the feed tray.

Figure 2.9.: The impact of the BT technique for component A: Initially, the bounds for the composition variables are 0 and 1. The light gray lines correspond to bounds obtained from the top-down propagation whereas the dark gray lines indicate the bounds from the bottom-up propagation.

Figure 2.9 (b) shows that imposing stronger bounds on V, $x_{\text{A},1}^{\text{rect}}$, and $x_{\text{A},1}^{\text{strip}}$ results in tighter bounds on the composition variables and a reduced range for the feed position indicated by the dashed lines. Besides that, the BT technique can be helpful to detect infeasible column designs very fast. For instance, Figure 2.9 (c) illustrates that if we reduce the length of the column sections from 75 trays to 60 trays, we get a conflict at the feed tray. Thus, the given range of specifications cannot lead to a feasible separation and can be excluded from the search space.

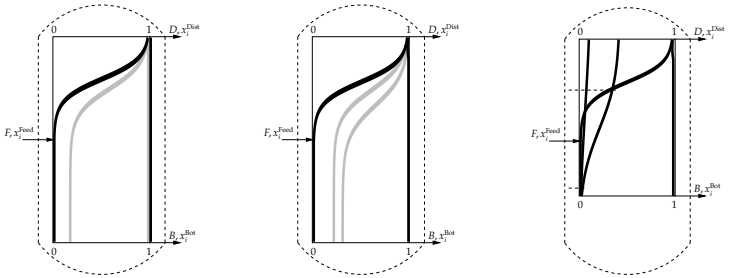
Comparison to BT by Interval Arithmetic The discussion above illustrates the potential of the proposed BT technique which we subsequently compare to standard BT. As displayed in Section 2.1 one of the most commonly used approaches is to use expression trees for the constraints together with interval arithmetic. For this, we can either use the material balance equations in Equation (2.14) or their equivalent representation in Equation (2.15). Depending on the formulation different

bounds are obtained. We observed that tighter bounds can be obtained via Equation (2.15). For $l = 1$ and $i = A$, Equation (2.15) evaluates to $y^{\text{rect}}_{A,2} = x^{\text{rect}}_{A,1} + \frac{D}{V}(x^{\text{Dist}}_{A} - x^{\text{rect}}_{A,1})$. As the expression is already solved for the desired variable $y^{\text{rect}}_{A,2}$, we can omit the expression tree and just apply interval arithmetic. Substituting each variable in $x^{\text{rect}}_{A,1} + \frac{D}{V}(x^{\text{Dist}}_{A} - x^{\text{rect}}_{A,1})$ by its interval from Table 2.2 we obtain $[0.98, 1] + [0.24, 0.25]/[2, 5]([0.99, 1] - [0.98, 1]) = [0.9815, 1]$ while our approach yields $y^{\text{rect}}_{A,2} \in [0.9834, 1]$ indicating a small advantage.

In order to reveal the strength of our approach, we further computed the bounds over the entire column. In Figures 2.10 (a) and (b) we show the boundary intervals obtained by the two methods for the rectifying section. The bold black lines correspond to bounds from interval arithmetic while the light gray lines depict bounds from our approach. In Figure 2.10 (b) the area between the boundary intervals from interval arithmetic is about 6.5 times larger than the area obtained by our BT technique. For ease of presentation we do not give the boundary intervals for the stripping section in Figures 2.10 (a) and (b). Note, however, that the bounds for the stripping section obtained from interval arithmetic are acceptable. They are 1.5 times worse than the bounds corresponding to our approach. Nevertheless, BT by interval arithmetic is not able to detect infeasibility when only 60 stages are active while our approach is able to do so. Compare Figure 2.10 (c) and Figure 2.9 (c), respectively. The horizontal dashed lines in Figure 2.10 (c) represent the area of possible feed positions.

Remark 2.6. We remark that shortcut methods (e.g., [LVD85, BWM98]) can also be used to compute bounds on some key variables, e.g., the minimal energy demand. These bounds reduce the overall domain but are usually rather weak when subdomains are investigated. This is a particular disadvantage for branch-and-bound based global optimization algorithms, where the domain is successively refined. The presented BT technique fits into this concept as it computes stronger bounds for smaller subdomains of $x^{\text{Dist}}_i, x^{\text{Bot}}_i, D, B, V$.

The next two subsections show how global optimization of distillation columns can benefit from the proposed BT technique.



(a) Parameters and domains according to Table 2.2.

(b) Stronger bounds: $3 \leq V \leq 4$, $x_{A,1} \leq 0.99$, $x_{A,N} \geq 0.005$

(c) Shorter Column: BT by interval arithmetic does not yield a conflict.

Figure 2.10.: Comparison between the results of a BT by interval arithmetic (bold black lines) and the proposed BT technique (light gray lines). In (a) and (b) only the rectifying section is considered. In (c) the boundary intervals for both sections are given according to BT by interval arithmetic.

2.3.2. A Relaxed MINLP Formulation

The illustration of the BT technique showed that BT does not only improve the bounds on the composition variables, but it can also indicate infeasibility of a certain range of variable specifications without using the complete model formulation. See Figure 2.9 (c). Therefore, we adopt the BT technique derived in the previous section such that it can be used as a search space reduction scheme.

Our approach extends the ideas of the McCabe-Thiele diagram and the *boundary value* method introduced by Levy et al. [LVD85], which can be summarized as follows. Given some *fixed* values for the product composition and the flow rates, the composition profiles of the rectifying and stripping section can be computed by tray-to-tray calculations using Formula (2.15). The separation is only feasible if the composition profiles intersect. The point of intersection is the feed tray. To determine the global optimal solution, all reflux ratios have to be checked (which is an infinite procedure due to the continuous domains).

We propose to analyze the column profiles not for fixed values of product compositions and flow rates but for intervals. Thus, extreme profiles

can be computed using the BT technique from the previous section which enclose all possible profiles. The extreme profiles lead to *boundary intervals* for the composition variables of the rectifying and stripping section. If the boundary intervals intersect for some choice of the column section lengths, the separation can be feasible. The point of intersection represents a possible feed tray (cf. Figures 2.9 (a) and (b)). If the column section lengths cannot be chosen such that the intervals intersect, there can be no successful separation for the given range of specifications (cf. Figure 2.9 (c)). Hence, the range of specifications can be excluded from the search space. These insights can be gained without the use of the material balance equations for the trays between the inlet and outlet trays.

We come up with the following smaller MINLP model which is a relaxation of a binary distillation column, and refer to it as relaxed MINLP model. It only consists of three material balance equations that correspond to the rectifying section, the feed tray, and the stripping section in order to model the incoming and outgoing streams. The course of the composition variables within the sections is relaxed by the boundary intervals instead of using all material balance equations.

$$\begin{aligned} Dx_i^{\mathrm{Dist}} &= Vy_{i,l^{\mathrm{F}}} - (V-D)x_{i,l^{\mathrm{F}}-1}, \\ -(Dx_i^{\mathrm{Dist}} + Bx_i^{\mathrm{Bot}}) &= Vy_{i,l^{\mathrm{F}}+1} - Vy_{i,l^{\mathrm{F}}}) + (V-D)x_{i,l^{\mathrm{F}}-1} \\ &\quad -(V+B)x_{i,l^{\mathrm{F}}}, \\ Bx_i^{\mathrm{Bot}} &= (V+B)x_{i,l^{\mathrm{F}}+1} - Vy_{i,l^{\mathrm{F}}}, \end{aligned} \tag{2.20}$$

with the phase equilibrium

$$y_{i,l} = \frac{\alpha_i x_{i,l}}{\alpha_A x_{\mathrm{A},l} + \alpha_B x_{\mathrm{B},l}}, \qquad l = l^{\mathrm{F}}, l^{\mathrm{F}}+1. \tag{2.21}$$

In order to couple the different sections with each other and also to associate the bounds obtained from BT with the variables, we introduce further the following coupling conditions

$$\begin{aligned} (\beta_l^{\mathrm{rect}} - 1) + \mathrm{lb}(x_{i,l}^{\mathrm{rect}}) &\leq x_{i,l^{\mathrm{F}}-1} \leq (1-\beta_l^{\mathrm{rect}}) + \mathrm{ub}(x_{i,l}^{\mathrm{rect}}), \\ (\beta_l^{\mathrm{rect}} - 1) + \mathrm{lb}(x_{i,l+1}^{\mathrm{rect}}) &\leq x_{i,l^{\mathrm{F}}} \leq (1-\beta_l^{\mathrm{rect}}) + \mathrm{ub}(x_{i,l+1}^{\mathrm{rect}}), \\ (\beta_l^{\mathrm{strip}} - 1) + \mathrm{lb}(x_{i,l+1}^{\mathrm{strip}}) &\leq x_{i,l^{\mathrm{F}}} \leq (1-\beta_l^{\mathrm{strip}}) + \mathrm{ub}(x_{i,l+1}^{\mathrm{strip}}), \\ (\beta_l^{\mathrm{strip}} - 1) + \mathrm{lb}(x_{i,l}^{\mathrm{strip}}) &\leq x_{i,l^{\mathrm{F}}+1} \leq (1-\beta_l^{\mathrm{strip}}) + \mathrm{ub}(x_{i,l}^{\mathrm{strip}}), \end{aligned} \tag{2.22}$$

where $\beta_l^{\text{rect}} \in \{0,1\}$, $l \in \{1,\dots,l_{\max}^{\text{rect}}\}$, is a binary variable which is active if and only if the length of the rectifying section is l, and $\beta_l^{\text{strip}} \in \{0,1\}$, $l \in \{1,\dots,l_{\max}^{\text{strip}}\}$, is a binary variable which is active if and only if the length of the rectifying section is l. To ensure that the lengths of the stripping and rectifying sections are uniquely determined, we demand

$$\sum_{l=1}^{l_{\max}^{\text{rect}}} \beta_l^{\text{rect}} = 1 \quad \text{and} \quad \sum_{l=1}^{l_{\max}^{\text{strip}}} \beta_l^{\text{strip}} = 1. \tag{2.23}$$

Finally, we bound the number of active trays from above by

$$\text{Number of active trays} = \sum_{l=1}^{l_{\max}^{\text{rect}}} \beta_l^{\text{rect}} l + \sum_{l=1}^{l_{\max}^{\text{strip}}} \beta_l^{\text{strip}} l + 1 \leq N_{\text{trays}}^{\max}. \tag{2.24}$$

For illustration, assume $\beta_s^{\text{rect}} = 1$ for $s \in \{1,\dots,l_{\max}^{\text{rect}}\}$ and $\beta_t^{\text{strip}} = 1$ for $t \in \{1,\dots,l_{\max}^{\text{strip}}\}$. Then, the sections are of length s and t, respectively, and the coupling condition (2.22) ensures that $\max\{\text{lb}(x_{i,s+1}^{\text{rect}}), \text{lb}(x_{i,t+1}^{\text{strip}})\} \leq x_{i,l^{\text{F}}} \leq \min\{\text{ub}(x_{i,l+1}^{\text{rect}}, \text{ub}(x_{i,s+1}^{\text{strip}}))\}$ which links the two sections. If it holds that $\max\{\text{lb}(x_{i,s+1}^{\text{rect}}), \text{lb}(x_{i,t+1}^{\text{strip}})\} > \min\{\text{ub}(x_{i,l+1}^{\text{rect}}, \text{ub}(x_{i,s+1}^{\text{strip}}))\}$, separation cannot be achieved.

The relaxed MINLP formulation for the distillation column allows us to define a relaxed MINLP model for the complete hybrid distillation/melt-crystallization processes by simply replacing the part of the distillation column by the formulas given in Equations (2.20) through (2.24) in the model. Note that the resulting relaxed model is still a mixed-integer nonlinear program. However, it can be solved more efficiently by available solvers due to its reduced problem size in terms of constraints and variables. If the relaxed model is infeasible, this proves that the corresponding subdomain cannot contain any feasible solution and can be excluded from the search space. Otherwise, the solution gives a lower bound on the minimization problem and helps to evaluate known local solutions. In Section 2.4 we make use of the relaxed MINLP model formulation within a comprehensive case study.

2.3.3. A Fixed Sections Modeling Approach

The relaxed MINLP model formulation can be extended to a complete MINLP model for distillation columns so that the improved bounds can be directly assigned to their corresponding variables yielding tighter convex relaxations. Note that this is not possible for the reference distillation column model (see Section 2.2), where the material conservation is modeled by the component material balance equations (2.3), i.e.,

$$(y_{i,l+1} - y_{i,l})\, V + x_{i,l-1} L_l - x_{i,l} L_{l+1} + x_i^{\mathrm{F}}\, F\, \beta_l^{\mathrm{F}} + x_i^{\mathrm{Dist}}\, R\, \beta_l^{\mathrm{Recycle}} = 0. \tag{2.25}$$

If a distillation column with fixed length and fixed feed position is considered, i.e., the values of the binary variables β_l^{F} and $\beta_l^{\mathrm{Recycle}}$ are fixed, it is clear where the stripping and rectifying sections start and end so that the improved bounds can be assigned directly to the variables. However, the length of a distillation column and the feed position are design parameters and are usually determined within the optimization process. Thus, assignment of a variable $x_{i,l}$ in Equation (2.25) to a specific tray in the stripping or rectifying section is highly dependent on the binary variables which makes it difficult to exploit this model structure for our BT technique. We extend the relaxed MINLP model such that the assignment of the variables to a column section is independent from the binary variables.

To complete the relaxed MINLP model, material balance equations are introduced for each tray of the rectifying and stripping section. The distillation column is divided in three independent, structurally constant sections, namely the rectifying, stripping, and feed section which are coupled by binary variables. The basic concept of such a *fixed sections modeling approach* (FSMA) is shown in Figure 2.11.

The mathematical description of the FSMA model is given by the material balance equations for the rectifying and stripping section

$$\begin{aligned} D\, x_i^{\mathrm{Dist}} &= V\, y_{i,l+1}^{\mathrm{rect}} - (V - D)\, x_{i,l}^{\mathrm{rect}}, \qquad l = 1, \ldots, l_{\max}^{\mathrm{rect}}, \\ B\, x_i^{\mathrm{Bot}} &= (V + B)\, x_{i,l+1}^{\mathrm{strip}} - V\, y_{i,l}^{\mathrm{strip}}, \qquad l = 1, \ldots, l_{\max}^{\mathrm{strip}}, \end{aligned} \tag{2.26}$$

the material balance equation for the feed section

$$\begin{aligned} & -(D x_i^{\mathrm{Dist}} + B x_i^{\mathrm{Bot}}) \\ = \;& V\, y_{i,l^{\mathrm{F}}+1} - V\, y_{i,l^{\mathrm{F}}} + (V - D)\, x_{i,l^{\mathrm{F}}-1} - (V + B)\, x_{i,l^{\mathrm{F}}}, \end{aligned} \tag{2.27}$$

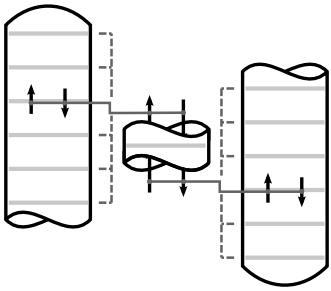


Figure 2.11.: Fixed sections modeling approach with rectifying section (left), feed section (middle), and stripping section (right) which can be coupled at variable positions. The feed stream and the product streams are not shown for better clarity of the figure.

the phase equilibrium corresponding to Equation (2.6), and the coupling conditions

$$\begin{array}{rcccl}
(\beta_l^{\text{rect}} - 1) + x_{i,l}^{\text{rect}} & \leq & x_{i,l^{\text{F}}-1} & \leq & (1 - \beta_l^{\text{rect}}) + x_{i,l}^{\text{rect}}, \\
(\beta_l^{\text{rect}} - 1) + x_{i,l+1}^{\text{rect}} & \leq & x_{i,l^{\text{F}}} & \leq & (1 - \beta_l^{\text{rect}}) + x_{i,l+1}^{\text{rect}}, \\
(\beta_l^{\text{strip}} - 1) + x_{i,l+1}^{\text{strip}} & \leq & x_{i,l^{\text{F}}} & \leq & (1 - \beta_l^{\text{strip}}) + x_{i,l+1}^{\text{strip}}, \\
(\beta_l^{\text{strip}} - 1) + x_{i,l}^{\text{strip}} & \leq & x_{i,l^{\text{F}}+1} & \leq & (1 - \beta_l^{\text{strip}}) + x_{i,l}^{\text{strip}}.
\end{array} \tag{2.28}$$

Moreover, Equations (2.23) and (2.24) regarding the binary variables are required as well as the following improved bounds on the composition variables

$$\begin{array}{rcccll}
\text{lb}(x_{i,l}^{\text{rect}}) & \leq & x_{i,l}^{\text{rect}} & \leq & \text{ub}(x_{i,l}^{\text{rect}}), & l = 1, \ldots, l_{\max}^{\text{strip}}, \\
\text{lb}(x_{i,l}^{\text{strip}}) & \leq & x_{i,l}^{\text{strip}} & \leq & \text{ub}(x_{i,l}^{\text{strip}}), & l = 1, \ldots, l_{\max}^{\text{strip}},
\end{array} \tag{2.29}$$

which are obtained by our BT technique in Equations (2.17) and (2.18). Note that Equations (2.28) and (2.29) imply the bounds on the variables $x_{i,l}$, $l \in \{l^{\text{F}} - 1, l^{\text{F}}, l^{\text{F}} + 1\}$.

We remark that the presented BT technique and its use in a relaxed MINLP formulation is not restricted to the presented FSMA model. It can,

for instance, also be used for the GDP model formulations introduced by Yeomans et al. [YG00].

2.4. A Case Study for Hybrid Distillation/Melt-Crystallization Processes

In this section we apply the presented approach to determine a cost-optimal process design of hybrid distillation/melt-crystallization processes for binary mixtures as introduced in Section 2.2. We begin with a brief description of our test instances in Section 2.4.1. In Section 2.4.2 we present three solution strategies. Two solution strategies are based on the different distillation column models, i.e., the reference and FSMA model, that are solved by state-of-the-art global optimization software. The third strategy makes use of the relaxed MINLP problem. Finally, we discuss the computational results obtained for each test instance by applying the solution strategies to both the stand-alone distillation processes (see Section 2.4.3) and the hybrid processes (see Section 2.4.4).

2.4.1. Test Instances

For our test instances we investigate the superstructure illustrated in Figure 2.6. It includes ten different process configurations (PC). Each process configuration consists of a distillation column that is combined with up to two crystallizers (see Figure 2.7). We consider this superstructure for fifteen different parameter sets. The specifications for our reference instance T0 are given in Table 2.3.

(x_A^{eut}, x_B^{eut})	(0.50,0.50)	(pur_A, pur_B)	(0.99,0.99)
(x_A^F, x_B^F)	(0.25,0.75)	(α_A, α_B)	(1.3,1.0)
F [mol s^{-1}]	1	N_{trays}^{max}	75
(k_1, k_2, k_3) [mol^{-1} s]	(1.0,6.00,50.0)		

Table 2.3.: Parameter specifications for the reference instance T0.

For each composition variable we assume the natural interval $[0, 1]$, while each variable representing a molar flow is bounded by $30\,\text{mol}\,\text{s}^{-1}$ from above. The further test instances are derived from the reference

instance T0 by varying one key parameter. They are described in Table 2.4.

Test series	Modified parameter	Instance a	Instance b
T1	(x_A^{eut}, x_B^{eut})	(0.25, 0.75)	(0.75, 0.25)
T2	(x_A^F, x_B^F)	(0.10, 0.90)	(0.40, 0.60)
T3	k_3 [mol^{-1} s]	60	75
T4	F [mol s^{-1}]	0.50	2.00
T5	(k_1, k_2) [mol^{-1} s]	(1.0, 0.6)	(1.0, 60.0)
T6	$(\mathrm{pur}_A, \mathrm{pur}_B)$	(0.90, 0.90)	(0.95, 0.95)
T7	(α_A, α_B) / N_{trays}^{max}	(1.15, 1.0) / 150	(1.45, 1.0) / 50

Table 2.4.: Specifications for further test instances.

2.4.2. Solution Strategies

In order to find the cost-optimal designs, we apply the following three solution strategies to each test instance.

(S1) The variable process structure is formulated as a MINLP, where the distillation column is modeled by the reference distillation column model (see Section 2.2). The resulting MINLP is then solved by the global optimization software BARON.

(S2) We use the FSMA model for the distillation column (see Section 2.3.3) and solve the MINLP by BARON.

(S3) The third solution strategy consists of two steps. In a first step, we apply a heuristic to each process configuration (time limit: 10 CPU seconds per each of the ten PC) in order to find feasible solutions. The objective function value of the best solution provides an upper bound on the optimal cost.

The second step is based on three subroutines. Given a subdomain $\mathcal{D}$, the first subroutine **Relaxed_Model**($\mathcal{D}$) constructs and solves the relaxed MINLP model over $\mathcal{D}$ (see Section 2.3.2). The second subroutine **Complete_Model**($\mathcal{D}$) uses the complete, but computationally expensive model on $\mathcal{D}$ where the distillation column is modeled using the FSMA formulation from Section 2.3.3. The two subroutines

ϵ	Δ	τ	$\mathtt{BF}(V)$	$\mathtt{BF}(D)$	$\mathtt{BF}(B)$	$\mathtt{BF}(x_i^{\mathrm{Dist}})$	$\mathtt{BF}(x_i^{\mathrm{Bot}})$
10^{-5}	0.50	0.25	1	1	1	4	4

Table 2.5.: Parameter specification for the modified branch-and-bound algorithm given in Algorithm 1 and used in solution strategy (S3).

are incorporated in a branch-and-bound framework that handles a successive refinement of the domains of x_i^{Dist}, x_i^{Bot}, V, B, and D. Here, subroutine **Relaxed_Model**($\mathcal{D}$) is used as long as the weighted interval length of the current branching variable is above a predetermined value τ. Otherwise, subroutine **Complete_Model**($\mathcal{D}$) is applied. The advantage of this combined method is that the first subroutine can detect infeasible or nonoptimal configurations and subdomains very fast. This reduces the number of calls of the computationally expensive second subroutine.

In addition, a third subroutine **Struct_Reduction**($\mathcal{D}$) is used if the weighted interval length of the current branching variable is below a pre-determined value Δ. This routine checks which of the fixed process configurations can still contain a global optimal solution over $\mathcal{D}$ and which of them can be already excluded from further considerations. For this, the binary variables indicating the existence of connections between the distillation column and the crystallizers (see Section 2.2) are minimized and maximized over the constraint set of the relaxed MINLP model. If possible, they are then fixed to either zero or one. Solution approach (S3) is formalized in Algorithm 1.

For our computations we use the parameter setting as given in Table 2.5 and the software package BARON to solve all subproblems constructed in the second step to global optimality. To find initial solutions in the first step of (S3), we apply the multi-start heuristic solver provided by BARON.

Discussion of Solution Strategy (S3) Solution strategy (S3) utilizes several ideas from BT (see Section 2.1). First, the BT technique presented in Section 2.3.1 is used in the Subroutines **Relaxed_Model**($\mathcal{D}$) and **Com-**

Algorithm 1 Modified B&B for hybrid separation processes.

Input: Specifications from Tables 2.3 and 2.4, the overall domain $\mathcal{D}^0 = [l, u]$, parameters $\Delta, \tau, \epsilon > 0$, and a branching factor $\texttt{BF(var)} > 0$ for each $\texttt{var} \in \texttt{VAR} := \{V, D, B, x_i^{\text{Dist}}, x_i^{\text{Bot}}\}$. $\mathcal{Q} := \{\mathcal{D}^0\}$.

1: **Local Search**: Apply a heuristic to each of the ten possible substructures separately for ten CPU seconds. Set UB to the objective function value corresponding to the best local solution (or to $+\infty$).
2: **Bound Tightening**: Min/max each variable from VAR over the relaxed model and update initial bounds (if possible).
3: **Root node relaxation**: Solve the relaxed MINLP over $\mathcal{D}^0$ and set $\texttt{LB}(\mathcal{D}^0)$ to its optimal objective function value (or to $+\infty$).
4: **if** $\texttt{LB}(\mathcal{D}^0) = +\infty$ **then**
5: **return** Problem is infeasible.
6: **else**
7: Set $\bar{\mathcal{D}} := \mathcal{D}^0$.
8: **while** $\texttt{UB} - \texttt{LB}(\bar{\mathcal{D}}) > \epsilon$ **do**
9: **Branching variable selection**: For all $\texttt{var} \in \texttt{VAR}$ compute the weighted interval length $L_{\texttt{var}} := (\text{ub}(\texttt{var}) - \text{lb}(\texttt{var})) \cdot \texttt{BF(var)}$. Choose $\overline{\texttt{var}} \in \texttt{VAR}$ with $L_{\overline{\texttt{var}}}$ maximal.
10: **Child nodes**: Bisect the interval of $\overline{\texttt{var}}$ yielding $\mathcal{D}'$ and $\mathcal{D}''$.
11: **for** $\mathcal{D} \in \{D', D''\}$ **do**
12: **if** $L_{\overline{\texttt{var}}} < \Delta$ **then**
13: **Struct_Reduction**($\mathcal{D}$): Min/Max all 0/1-variables determining the superstructure to fix their values (relaxed model).
14: **if** $L_{\overline{\texttt{var}}} < \tau$ **then**
15: **Complete_Model**($\mathcal{D}$): Solve the complete model over $\mathcal{D}$ and set $\texttt{LB}(\mathcal{D})$ to the optimal obj. func. value (or to $+\infty$).
16: **if** $\texttt{UB} - \texttt{LB}(\mathcal{D}) > \epsilon$ **then**
17: Set $\texttt{UB} = \texttt{LB}(\mathcal{D})$.
18: **else**
19: **Relaxed_Model**($\mathcal{D}$): Solve the relaxed MINLP and set $\texttt{LB}(\mathcal{D})$ to the optimal objective function value (or to $+\infty$).
20: **if** $\texttt{UB} - \texttt{LB}(\mathcal{D}) > \epsilon$ **then**
21: $\mathcal{Q} := \mathcal{Q} \bigcup \{\mathcal{D}\}$
22: $\mathcal{Q} := \mathcal{Q} \setminus \{\bar{\mathcal{D}}\}$.
23: **Look up** $\bar{\mathcal{D}} \in \mathcal{Q}$ **with the worst lower bound** $\texttt{LB}(\bar{\mathcal{D}})$
24: **return** UB

plete_Model($\mathcal{D}$) to determine the bounds on the composition variables associated to the relaxed MINLP model for the distillation column and the FSMA model, respectively. Second, certain key variables are minimized and maximized over a relaxed model formulation which is a known concept in the BT area. Third, the objective function values corresponding to the feasible solutions are used as upper bounds on the objective function. Thus, all further BT techniques may exclude feasible operating points which fits into the concept of OBBT.

Computational Settings Solution strategies (S1) and (S2) use the software BARON 9.0.7 [TS05] (with default settings, CPLEX as LP-subsolver and SNOPT as NLP-subsolver) in the GAMS 23.6.2 environment [GAM09]. Solution strategy (S3) as displayed in Algorithm 1 was implemented in the programming language C. All subproblems constructed in (S3) are modeled with SCIP 1.2.0 [Ach07] and solved by BARON 9.0.7 [TS05] (with default settings, CPLEX as LP-subsolver and SNOPT as NLP-subsolver).

Remarkably our local search heuristic in (S3), where each of the ten possible process configurations is considered separately, computes often better initial solution than (S1) and (S2). Note that good feasible solutions have a potential impact on the generation of lower bounds on the problem. Therefore, we also provide the initial solution found by (S3) to (S1) and (S2) in a second test run. We observed that both runs almost lead to the same result for (S1) and (S2). The symbol '*' in the subsequent tables indicates that the better lower bound was obtained using the provided initial solution.

All computations are carried out on a 2.67 GHz INTEL X5650 with 96 GB RAM and are stopped, if necessary, after 100:00 hours.

Remark 2.7. We further tested modifications of the distillation column models used in (S1) and (S2). For (S1) we simplified some nonlinearities in the reference distillation column model using a Big-M formulation. For (S2) we reduced the number of binary variables in the FSMA distillation column model using the modeling technique of Vielma and Nemhauser [VN11]. Some preliminary tests of the two modifications showed similar or worse performance so that we decided to work with the two presented model formulations (see Table A.1 in Appendix A).

2.4.3. Computational Results for a Distillation Column

The computational results of our solution strategies applied to a stand-alone distillation column are summarized in Table 2.6. Note that the results for the test instances T1a, T1b, T3a, and T3b are neglected as the design problems for the underlying stand-alone distillation columns are all identical to the one of our reference instance T0. This is due to the fact that the parameters specifying these test instances only concern the crystallizers. We can first observe that solution strategy (S2) cannot solve

	Opt. cost	Lower bounds and CPU time (hh:mm) (S1)	(S2)	(S3)	Initial cost (S3)
T0	**306.3**	255.0 (100:00)	243.5* (100:00)	**306.3** (00:06)	308.5
T2a	**254.3**	228.6 (100:00)	235.3 (100:00)	**254.3** (00:07)	∞
T2b	**326.1**	281.1 (100:00)	232.5 (100:00)	**326.1** (00:06)	326.1
T4a	**153.1**	130.0 (100:00)	111.2* (100:00)	**153.1** (00:07)	154.2
T4b	**612.7**	524.3 (100:00)	506.4 (100:00)	**612.7** (00:11)	∞
T5a	**282.0**	255.1 (100:00)	223.3* (100:00)	**282.0** (00:07)	285.4
T5b	**531.1**	448.9* (100:00)	425.7* (100:00)	**531.1** (00:04)	531.1
T6a	**106.2**	89.2 (100:00)	74.1 (100:00)	**106.2** (00:06)	106.2
T6b	**175.1**	152.5 (100:00)	128.4* (100:00)	**175.1** (00:05)	175.1
T7a	**1056.4**	227.3 (100:00)	638.8 (100:00)	**1056.4** (02:51)	∞
T7b	**155.5**	**155.5** (94:35)	132.8 (100:00)	**155.5** (00:02)	156.3

Table 2.6.: Stand-alone distillation column: Optimal cost in comparison with the lower bounds of the solution strategies (S1) to (S3) after 100 hours or the time needed to solve the problem globally. The last column provides the initial cost determined in the first step of (S3) or ∞ if no feasible solution was found. Bounds that are labeled by the symbol '*' in Column 3 and 4 are obtained by the second test run that uses the initial cost of (S3).

the underlying MINLP within the time limit of 100:00 hours and (S1) is only able to guarantee global optimality of test instance T7b after 94:35 hours. Solution strategy (S3) can solve all but one test instance globally within a few minutes.

For (S3) there are two important issues that need to be discussed, namely the impact of the initial cost of (S3) and the problem size of test instances T7a and T7b. For test instances T2a, T4b, and T7a, no local solution is found by the preprocessing step of (S3) which requires to set the initial cost to infinity. In these cases (S3) needs more computation

time. In fact, if we provide the optimal solutions as the local solutions, the computation time reduces to 00:05 hours, 00:07 hours and 00:33 hours, for T2a, T4b, and T7a, respectively.

Another reason for the large computation time of (S3) to solve test instance T7a is due to the large underlying problem. Recall that for test instance T7a the constant relative volatilities (α_A, α_B) have been changed from $(1.3, 1)$ to $(1.15, 1)$, which requires to consider a larger distillation column for which the maximum number of trays is increased from 75 to 150. Thus, the problem size of the corresponding MINLP is significantly larger. In addition, the higher the number of possible trays, the worse becomes the quality of the improved bounds obtained by our tray-to-tray calculation (see Formulas (2.17) and (2.18)). This may also explain why test instance T7b, that includes a distillation column with a length of at most 50 trays, can be solved much faster.

Summing up, both (S1) and (S2) cannot determine the cost-optimal design of a stand-alone distillation column in reasonable time while solution strategy (S3) is able to solve the design problem and to reduce the computation time by orders of magnitude. It benefits heavily from the techniques introduced in Section 2.3.

2.4.4. Computational Results for Hybrid Processes

Our computational results for hybrid distillation/melt-crystallization processes are summarized in Table 2.7 and Table 2.8. The results in Table 2.7 show that the underlying MINLPs of solution strategies (S1) and (S2) cannot be solved within 100:00 hours by BARON. Thus, both strategies only provide lower bounds on the optimal cost which, in addition, are extremely poor. Using the modified branch-and-bound algorithm of solution strategy (S3), we are able to solve all but test instance T7a to global optimality within at most 66 hours. To solve T7a to global optimality, our solution approach (S3) needs a computation time of 113:42 hours. After the time limit of 100:00 hours (S3) nevertheless computes a lower bound of 233.25 for T7a which clearly outperforms the bounds 9.83 and 26.12 obtained by (S1) and (S2). Again, this test instance is computationally more expensive than the other test instances due to its larger problem size. Furthermore, the gap between its initial solution of 324.35 and its optimal solution of 239.22 is much larger compared to the other test instances. In fact, if we use the optimal solution as starting local solution,

	Opt. cost	Lower bounds and CPU time (hh:mm) (S1)	(S2)	(S3)	Initial cost (S3)
T0	**154.0**	12.6 (100:00)	57.2 (100:00)	**154.0** (26:17)	154.0
T1a	**203.3**	16.2* (100:00)	51.7* (100:00)	**203.3** (55:53)	203.3
T1b	**120.8**	11.4* (100:00)	58.3* (100:00)	**120.8** (60:00)	142.9
T2a	**86.7**	3.9 (100:00)	24.7* (100:00)	**86.7** (17:15)	86.7
T2b	**203.5**	17.9 (100:00)	61.0* (100:00)	**203.5** (34:00)	203.5
T3a	**173.4**	14.1 (100:00)	57.2* (100:00)	**173.4** (23:37)	173.4
T3b	**201.1**	11.5* (100:00)	50.2* (100:00)	**201.1** (32:06)	201.1
T4a	**77.0**	6.3* (100:00)	7.3* (100:00)	**77.0** (43:03)	82.5
T4b	**308.0**	24.8 (100:00)	172.6 (100:00)	**308.0** (46:21)	308.0
T5a	**140.4**	28.1* (100:00)	53.1 (100:00)	**140.4** (19:56)	140.4
T5b	**271.3**	66.9* (100:00)	88.3 (100:00)	**271.3** (65:52)	281.2
T6a	**93.6**	11.0 (100:00)	16.2 (100:00)	**93.6** (2:37)	93.6
T6b	**127.4**	2.1 (100:00)	43.9 (100:00)	**127.4** (8:00)	127.4
T7a	**239.2**	9.8* (100:00)	26.1* (100:00)	233.2 (100:00)	324.3
T7b	**120.7**	31.1* (100:00)	56.1* (100:00)	**120.7** (9:40)	126.4

Table 2.7.: Hybrid processes: Optimal cost in comparison with the lower bounds of the solution strategies (S1) to (S3) after 100 hours or the time needed to solve the problem globally. The last column provides the initial cost determined in the first step of (S3). Bounds labeled by the symbol '*' in Column 3 and 4 are obtained by the second test run that uses the initial cost of (S3).

the computation time of strategy (S3) decreases to 101:49 hours.

It is noticeable that the computation time of (S3) varies significantly for the different test instances. To understand these large deviations, we focus on the optimal process configurations given in Column 2 of Table 2.8. We observe that the computationally less expensive test instances T2a, T6a, T6b, and T7b lead to the optimal process configurations PC05 or PC09 while the computationally expensive test instances T1a, T1b, T4a, T4b, T5b, and T7a lead to optimal process configurations PC04 or PC10.

Hence, we conjecture that the optimal process configuration has a significant impact on the computation time of (S3). To give evidence for this, we additionally solved the models corresponding to each fixed process configuration of the reference instance T0. These models are derived from the model of the variable process structure by simply fixing the binary

	Process config.	Distillation column Length/ Feed pos.	Vapor flow $\mathrm{mol\,s^{-1}}$	Condenser reflux $\mathrm{mol\,s^{-1}}$	Feed of crystallizer A $\mathrm{mol\,s^{-1}}$	B $\mathrm{mol\,s^{-1}}$
T0	PC10	17/8	2.4091	1.9165	0.4926	1.4791
T1a	PC04	17/8	4.6095	3.9240	0.6854	1.2612
T1b	PC10	16/8	1.7057	1.2763	0.4294	1.2379
T2a	PC09	25/8	0.8955	0.7159	0.1796	1.0000
T2b	PC04	17/10	4.6330	3.8763	0.7566	1.1825
T3a	PC10	18/9	2.4424	1.9879	0.4545	1.4598
T3b	PC09	34/10	2.3829	1.9720	0.4109	1.0000
T4a	PC10	17/8	1.2111	0.9677	0.2434	0.7394
T4b	PC10	17/8	4.8154	3.8269	0.9885	2.9563
T5a	PC10	15/7	2.6349	2.1309	0.5039	1.4829
T5b	PC10	22/10	2.0587	1.5206	0.5381	1.5123
T6a	PC05	24/16	1.4545	1.2670	0	1.0000
T6b	PC09	21/7	2.0932	1.6755	0.4177	1.0000
T7a	PC10	17/8	4.5668	3.7107	0.8562	1.8275
T7b	PC09	26/8	1.6222	1.2465	0.3757	1.0000

Table 2.8.: Characteristics of the optimal operating points found by solution strategy (S3).

variables indicating the connections between the distillation column and the crystallizers. For each process configuration we bound the cost function of the underlying model in (S3) by the globally optimal cost. For solution strategies (S1) and (S2) we solved each MINLP twice, with and without bounding the cost function by the globally optimal cost. The results are shown in Table 2.9. Again, the MINLP models cannot be solved within our computation time limit of 100:00 hours when solution strategy (S1) or (S2) is used.

Solution strategy (S3) can solve each problem in less than 14:05 hours of computation time when we use the corresponding optimal solutions as initial cost for (S3). We can, however, observe that the computation time differs significantly for different types of process configurations. The hardest problems leading to computation times between 05:12 hours and 14:05 hours correspond to process configurations where two crystallizers are involved and both output streams of the distillation column are fed to

	Opt. cost	Lower bounds and CPU time (hh:mm) (S1)	(S2)	(S3)	Initial cost (S3)
PC01	**306.3**	256.3 (100:00)	247.1* (100:00)	**306.3** (00:05)	306.3
PC02	**213.8**	55.4 (100:00)	126.7 (100:00)	**213.8** (00:31)	213.8
PC03	**265.6**	46.9 (100:00)	84.8* (100:00)	**265.6** (01:04)	265.6
PC04	**181.8**	70.9 (100:00)	86.2 (100:00)	**181.8** (07:46)	181.8
PC05	**197.2**	166.3 (100:00)	141.1* (100:00)	**197.2** (00:06)	197.2
PC06	**187.4**	87.6 (100:00)	111.0* (100:00)	**187.4** (00:44)	187.4
PC07	**452.2**	143.8 (100:00)	243.8 (100:00)	**452.2** (01:03)	452.2
PC08	**362.0**	153.7 (100:00)	250.4 (100:00)	**362.0** (14:05)	362.0
PC09	**165.0**	83.0 (100:00)	85.4 (100:00)	**165.0** (00:35)	165.0
PC10	**154.0**	86.1* (100:00)	96.3 (100:00)	**154.0** (05:12)	154.0

Table 2.9.: Single process configurations of T0: Optimal cost in comparison with the lower bounds of (S1) to (S3) after 100 hours or the time needed to solve the problem globally. Bounds that are labeled wit '*' in are obtained by the second test run that uses the initial cost of (S3). The optimal cost is used as initial cost.

the crystallizers, i.e., PC04, PC08, PC10. The second type of process configurations is characterized by the fact that one component is withdrawn from one of the outlet trays of the distillation column while the second output stream is fed into a crystallizer (PC02, PC03, PC06, PC07, PC09). The computation time for problems underlying this type of process configurations ranges from 31 minutes to 1:05 hours. The easiest problems with a computation time of five and six minutes are given by the process configurations PC01 and PC05, respectively. The two process configurations have in common that both components A and B are withdrawn at the outlet trays of the distillation column.

The different computation times can be explained by the characteristics of the BT technique introduced in Section 2.3.1. If a column outlet also forms an overall system outlet, the purity requirement conditions provide tight bounds on the composition variables corresponding to the outlet tray. These bounds are then propagated through the column section leading to tighter bounds on the other composition variables of the distillation column and hence to a stronger relaxed MINLP formulation (cf. Section 2.3.2). On the other hand, if a column outlet is fed back into the system then the bounds obtained by the BT technique can be rather weak. Thus, we can conclude that the modified branch-and-bound algorithm of

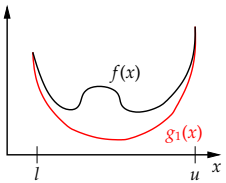
our solution strategy (S3) works best when the BT technique proposed in Section 2.3.1 can exploit tight bounds on the outlet trays of the distillation column.

This section showed that the global optimization techniques developed in Section 2.3 and applied in (S3) are reliable tools to detect the optimal design of hybrid distillation/melt-crystallization processes. Solution strategy (S3) determines the globally optimal design or at least strong bounds, while the bounds of solution strategies (S1) and (S2) which are based on standard optimization software are very poor. The average gap between the bounds of (S3) and the best bound of (S1) and (S2) is about 290%.

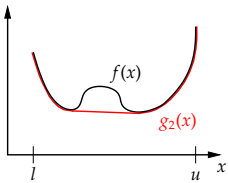
CHAPTER 3

Underestimation of Bivariate Functions

The quality of a convex relaxation of the graph of a function does not only depend on the underlying domain as discussed in the previous chapter but also on the specific under- and overestimators of the function. In Figure 3.1 two different convex underestimators of a function f over a domain $[l, u]$ are compared. The underestimator in Figure 3.1 (b) is the best possible convex underestimator of the function f over the domain $[l, u]$ – the so-called *convex envelope*.



(a) Convex underestimator $g_1(x)$.



(b) Convex envelope $g_2(x)$.

Figure 3.1.: Convex underestimators of a function $f(x)$ over $[l, u]$.

In Section 3.1 we discuss the fundamental concepts of convex envelopes. In particular, we emphasize that the evaluation of the convex

envelopes at a given point generally requires solving a highly nonconvex problem. We review some classes of functions whose structure can be exploited to determine the convex envelope. Nevertheless, there are still many functions whose convex envelope is not known, even in the uni- and bivariate case. Motivated by this, we implement a *cut-generation algorithm* for bivariate functions with certain structural properties in Section 3.2 to compute their convex envelopes numerically. This section is based on [BMV13].

Practical applications usually involve functions for which the envelopes are not at hand. In such cases it is important to find strong alternative relaxations. This is essential both from a practical and theoretical point of view as one can often find similar structures in different problems. In Section 3.3 we investigate a *chromatographic separation process* involving *second-order isotherms* which are functions of the form

$$f(x_1, x_2) = \frac{q_s x_1 (b_{1,0} + 2b_{2,0}x_1 + b_{1,1}x_2)}{1 + b_{1,0}x_1 + b_{0,1}x_2 + b_{2,0}x_1^2 + b_{1,1}x_1x_2 + b_{0,2}x_2^2} \tag{3.1}$$

with $q_S, b_{1,0}, b_{0,1}, b_{2,0}, b_{1,1}, b_{0,2} \geq 0$. The convex envelope of this function is not known and the function does not fit into our cut-generation algorithm. We analyze several relaxation strategies for second-order isotherms based on the concepts presented in Section 3.1. This section is an extension of [BMSMW10].

3.1. Convex Envelopes

Closed-form expressions for convex envelopes are only known for particular classes of functions over particular domains. In this work we mainly focus on the most common domain for global optimization issues, namely boxes $D = [l, u] \subseteq \mathbf{R}^n$. Over these box domains the functions, for which convex envelopes are available, can be partitioned into functions with polyhedral and nonpolyhedral convex envelopes (cf. [KS12a]). Functions with polyhedral convex envelopes include bilinear [McC76], trilinear [MF03, MF04], three dimensional edge-concave [MF05], submodular and edge-concave [TRX12], and some classes of multilinear functions [She97]. Nonpolyhedral convex envelopes are known for fractional terms x/y [TS01], some classes of (n-1)-convex functions with indefinite Hessian [JMW08], functions whose convex envelopes are generated by pairwise complementary convex combinations [Taw10], and functions

which are the product of convex and component-wise concave functions [KS12a, KS12b].

After a general introduction to the concept of convex envelopes, we focus on functions with polyhedral convex envelopes in Section 3.1.1, (n-1)-convex functions with indefinite Hessian in Section 3.1.2, and products of convex and component-wise concave functions in Section 3.1.3. Explicit convex envelopes are given for functions which occur in the application in Section 3.3.

We start with a formal definition of *convex* and *concave envelopes*.

Definition 3.1 ([HT96])**.** Let $D \subseteq \mathbf{R}^n$ be a convex, compact subset and $f : D \to \mathbf{R}$ be a real-valued function. The tightest convex underestimator of f over D is called the *convex envelope*, denoted by $\mathrm{vex}_D[f]$, while the tightest concave overestimator of f over D is called *concave envelope*, denoted by $\mathrm{cave}_D[f]$. The envelopes are defined pointwise:

$$\begin{aligned}\mathrm{vex}_D[f](x) = \max\{\eta(x) \mid \eta : D \to \mathbf{R} \cup \{\pm\infty\} \text{ with} \\ \eta(x) \leq f(x) \text{ for all } x \in D, \text{ and } \eta \text{ convex}\}, \\ \mathrm{cave}_D[f](x) = \min\{\eta(x) \mid \eta : D \to \mathbf{R} \cup \{\pm\infty\} \text{ with} \\ \eta(x) \geq f(x) \text{ for all } x \in D, \text{ and } \eta \text{ concave}\}.\end{aligned}$$

Remark 3.2. In general, we focus on convex envelopes since the same arguments can be applied for the concave envelope using the relation $\mathrm{vex}_D[f] = -\mathrm{cave}_D[-f]$.

The definition of the convex envelope does not give any suggestion for its computation. Another more constructive characterization can be obtained by the following definitions:

Definition 3.3.

1. Consider a function $f : D \to \mathbf{R}$ with $D \subseteq \mathbf{R}^n$. The *epigraph* of a function f over a domain $D \subseteq \mathbf{R}^n$ is defined as $\mathrm{epi}_D[f] = \{(x, \mu) \in \mathbf{R}^{n+1} \mid \mu \geq f(x) \quad \forall\, x \in D\}$.

2. The *convex hull* of a set $M \subseteq \mathbf{R}^n$ is defined as

$$\mathrm{conv}(M) = \left\{ x \in \mathbf{R}^n \mid x = \sum_k \lambda_k x^k, \lambda_k \geq 0, \sum_k \lambda_k = 1 \text{ and } x^k \in M \quad \forall\, k \right\}.$$

With these definitions, the convex envelope of a function f can be represented as a kind of "minimization" problem, as stated in [Roc70, HT96]:

$$\mathrm{vex}_D[f](x) = \inf\left\{\mu \mid (x,\mu) \in \mathrm{conv}(\mathrm{epi}_D[f])\right\}. \tag{3.2}$$

Subsequently, we assume that $f : D \subseteq \mathbf{R}^n \to \mathbf{R}$ is a *continuous* function and the domain D is a polytope. These assumptions reflect the general setting of most optimization problems. As the domain D is compact and f is continuous, the *Extreme Value Theorem* implies that the infimum of Problem (3.2) is attained at a point $x \in D$. Therefore, the convex envelope of a function at a given point x can be computed by the following nonconvex optimization problem

$$\begin{array}{lrcl} \min & \sum_k \lambda_k f(x^k) & & \\ \text{s.t.} & x & = & \sum_k \lambda_k x^k, \\ & 1 & = & \sum_k \lambda_k, \\ & \lambda_k \geq 0, & x^k \in D, & \text{for all } k. \end{array} \tag{VEX}$$

In order to simplify (VEX), two aspects of this problem can be considered: (i) An upper bound on the number of summands and (ii) a subset $\tilde{D}$ of D with $\mathrm{vex}_D[f](x) = \mathrm{vex}_{\tilde{D}}[f](x)$ for all $x \in D$. A natural upper bound on the number of summands is given by Carathéodory's Theorem (cf. [Roc70]). As the point $(x, \mathrm{vex}_D[f](x))$ is an element in the boundary of the convex set $\mathrm{conv}(\mathrm{epi}_D[f]) \subseteq \mathbf{R}^{n+1}$, we are in an n dimensional subspace and thus, the point can be written as convex combination of at most $n+1$ points. The minimal subset $\tilde{D}$ of D with $\mathrm{vex}_D[f](x) = \mathrm{vex}_{\tilde{D}}[f](x)$ for all $x \in D$ is called the *generating set* of $\mathrm{vex}_D[f](x)$.

Definition 3.4 ([Rik97])**.** Let $f : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a continuous function on a convex, compact domain D. Then, the *generating set of the convex envelope* of f over D is defined as

$$G_D^{\mathrm{vex}}[f] = \left\{x \mid (x, \mathrm{vex}_D[f](x)) \text{ is an extreme point of } \mathrm{conv}(\mathrm{epi}_D[f])\right\}.$$

A sufficient condition that helps to determine whether a point $x \in D$ does not belong to the generating set $G_D^{\mathrm{vex}}[f]$ is given by the next statement.

Observation 3.5 ([TS02a])**.** *Let $f : \mathbf{R}^n \to \mathbf{R}$ be restricted to a convex, compact*

subset $D \subseteq \mathbf{R}^n$. If there is a line segment $s \subseteq D$ such that $x \in D$ is contained in the relative interior ri(s) and f is concave over ri(s), then $x \notin G_D^{\text{vex}}[f]$.

The convex envelope over a domain D can be related to the convex envelope over a face of D. This allows us to consider lower dimensional spaces in order to investigate the generating set, for example.

Observation 3.6 ([TS02a])**.** *Let $f : \mathbf{R}^n \to \mathbf{R}$ be restricted to a convex, compact subset $D \subseteq \mathbf{R}^n$. Consider a nonempty face D' of D. Then,* $\text{vex}_D[f](x) = \text{vex}_{D'}[f](x)$ *for all $x \in D'$.*

Example 3.7. Let $f(x, y) := x^{0.3}y^2$ be restricted to $D := [1, 2] \times [3, 4]$. The function is concave along each segment connecting the points (l_x, y) and (u_x, y), $y \in [3, 4]$. Observation 3.5 implies $G_D^{\text{vex}}[f] \subseteq \{1, 2\} \times [3, 4]$. The function f is strictly convex restricted to the faces $D' = \{1\} \times [3, 4]$ and $D'' = \{2\} \times [3, 4]$. The generating set of strictly convex functions over a domain equals the domain. By Observation 3.6, $G_D^{\text{vex}}[f] = \{1, 2\} \times [3, 4]$. ⋄

3.1.1. Polyhedral Convex Envelopes

Functions with a polyhedral convex envelope are intensively studied in the literature as they exhibit nice combinatorial properties due to the character of their generating set.

Definition 3.8 ([Tar03, Tar08])**.** Let $f : D \to \mathbf{R}$, where $D \subseteq \mathbf{R}^n$ is a polytope. The convex envelope of f is called *polyhedral* if its generating set $G_D^{\text{vex}}[f]$ is finite. It is called *vertex polyhedral* if $G_D^{\text{vex}}[f] = \text{vert}(D)$.

Equivalently, polyhedral convex envelope can be defined to be the maximum of a finite collection of affine functions. For continuously differentiable functions it is sufficient to consider the vertices of D in order to check if the convex envelope is polyhedral.

Theorem 3.9 ([Rik97])**.** *Let $f : D \to \mathbf{R}$ be a continuously differentiable function, where $D \subseteq \mathbf{R}^n$ is a polytope. The convex envelope of f is polyhedral if and only if it is vertex polyhedral.*

There is no necessary condition on functions which guarantees polyhedral convex envelopes and is easy to check (cf. [Tar08]). The following class of functions satisfies, however, a sufficient condition for polyhedral convex envelopes.

Definition 3.10 ([Tar03])**.** Let $f : D \to \mathbf{R}$, where $D \subseteq \mathbf{R}^n$ is a polytope. The function f is called *edge-concave* over D if f is concave along all directions parallel to the edges of D.

Functions like $-x^2y^3$ restricted to boxes $[l, u] \subseteq \mathbf{R}^2_{\geq 0}$ belong to this class. Observation 3.5 implies the next result.

Theorem 3.11 ([Tar03])**.** *Let $f : D \to \mathbf{R}$, where $D \subseteq \mathbf{R}^n$ is a polytope. If f is edge-concave on D, then $\operatorname{vex}_D[f]$ is vertex polyhedral.*

The domain D is always assumed to be a box $[l, u]$ in the remainder of this chapter. In this case edge-concave functions are equivalent to *component-wise* concave functions, i.e., $f(x)$ is concave in x_i, $i = 1, \ldots, n$, for all fixed values of $x_j \in [l_j, u_j]$, $j \in \{1, \ldots, n\}$, $j \neq i$. The next example verifies that Theorem 3.11 provides only a sufficient condition for vertex polyhedral convex envelopes.

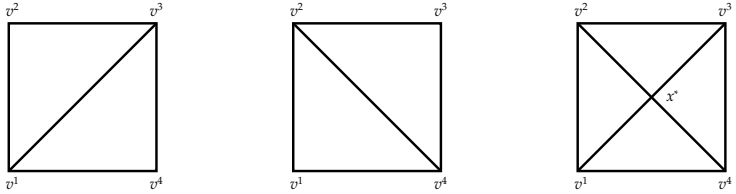
Example 3.12. Consider $f(x) := x^3$ restricted to $D = [-2, 1]$. This function is not component-wise concave in x. Yet, its convex envelope is polyhedral and given by $\operatorname{vex}_D[f] = 3x - 2$. ⋄

The evaluation of polyhedral convex envelopes at a given point $\bar{x}$, i.e., problem (VEX), and the determination of the corresponding affine function defining the convex envelope are the primal and dual version of a linear program (see [BST09, TRX12]):

$$P(\bar{x}): \quad \begin{array}{ll} \min_\lambda & f(V)^\intercal \lambda \\ \text{s. t.} & V\lambda = \bar{x},\ e^\intercal \lambda = 1, \\ & \lambda \geq 0, \end{array} \qquad D(\bar{x}): \quad \begin{array}{ll} \max_{(a,b)} & a^\intercal \bar{x} + b \\ \text{s. t.} & V^\intercal a + be \leq f(V), \\ & (a, b) \in \mathbf{R}^{n+1}, \end{array}$$

where $V = (v^1, \ldots, v^{2^n})$ corresponds to the set of vertices of a given box $[l, u] \subseteq \mathbf{R}^n$, and $f(V) = (f(v^1), \ldots, f(v^{2^n}))^\intercal$. If the affine function $a^\intercal x + b$ is irredundant in the description of $\operatorname{vex}_D[f]$, then (a, b) is the optimal solution of the dual program $D(x)$ for all x of a specific polyhedral subdomain of $[l, u]$. According to [TRX12] each subdomain can be refined into a triangulation by simplices. The union of such simplices over all irredundant affine functions defining the convex envelope forms a triangulation of the box. Therefore, *a vertex polyhedral convex envelope over a box corresponds to a certain triangulation of the box*. This allows to use combinatorial software to "enumerate" all triangulations and to determine the convex envelope, e.g., `QHull` [BDH96], `PORTA` [CL07], or `polymake` [GJ00]. However, such an approach is too expensive to be incorporated into global optimization software. See Section 4.4, for computational evidence.

Closed-form expressions for the convex envelope of component-wise concave functions or procedures to efficiently compute them are known up to dimension three. The idea is to determine the triangulation of the box which is associated with the convex envelope. Hence, the more possible triangulations the harder the computation of the convex envelope. In dimension one the function is concave and there is only one possible triangulation so that the convex envelope is simply the secant of the graph of the function (cf. [FS69]). In dimension two there are two possible triangulations of an arbitrary box whose vertices are denoted by v^1, v^2, v^3, and v^4 as indicated in Figure 3.2. The two triangulations are $T_1 = \{\{v^1, v^2, v^3\}, \{v^1, v^3, v^4\}\}$ and $T_2 = \{\{v^1, v^2, v^4\}, \{v^2, v^3, v^4\}\}$, which are given in Subfigures 3.2 (a) and (b), respectively. To determine the triangulation corresponding to the convex envelope of f, the two possible underestimators can be compared at the midpoint $x^* = \frac{1}{2}v^1 + \frac{1}{2}v^3 = \frac{1}{2}v^2 + \frac{1}{2}v^4$ of the box, cf. Subfigure 3.2 (c). If $\frac{1}{2}f(v^1) + \frac{1}{2}f(v^3) \leq \frac{1}{2}f(v^2) + \frac{1}{2}f(v^4)$, then T_1 generates the convex envelope. Otherwise, it is generated by T_2.



(a) Triangulations T_1.

(b) Triangulations T_2.

(c) Intersection.

Figure 3.2.: Triangulations of a two dimensional box.

The above considerations lead to the following closed-form expressions for vertex polyhedral convex envelopes of bivariate functions.

Proposition 3.13 (cf. [McC76, Tar03, Ben04, KS12a])**.** *Let $f : [l, u] \subseteq \mathbf{R}^2 \to \mathbf{R}$, $(x, y) \mapsto f(x, y)$, be a function with a vertex polyhedral convex envelope. If $f(l_x, l_y) + f(u_x, u_y) \leq f(l_x, u_y) + f(u_x, l_y)$, then*

$$\operatorname{vex}_{[l,u]}[f](x, y) = \begin{cases} \alpha_1 x + \beta_1 y + \gamma_1, & \text{if } y_0 \leq \frac{u_y - l_y}{u_x - l_x}(x_0 - l_x) + l_y, \\ \alpha_2 x + \beta_2 y + \gamma_2, & \text{if } y_0 > \frac{u_y - l_y}{u_x - l_x}(x_0 - l_x) + l_y, \end{cases}$$

where

$$\begin{aligned}
\alpha_1 &= \tfrac{f(u_x,l_y)-f(l_x,l_y)}{u_x-l_x}, \quad \beta_1 = \tfrac{f(u_x,u_y)-f(u_x,l_y)}{u_y-l_y},\\
\gamma_1 &= \tfrac{u_x(u_y-l_y)f(l_x,l_y)-(u_x-l_x)l_y f(u_x,u_y)+(u_x l_y-l_x u_y)f(u_x,l_y)}{(u_x-l_x)(u_y-l_y)} \quad ,\\
\alpha_2 &= \tfrac{f(u_x,u_y)-f(l_x,u_y)}{u_x-l_x}, \quad \beta_2 = \tfrac{f(l_x,u_y)-f(l_x,l_y)}{u_y-l_y},\\
\gamma_2 &= \tfrac{(u_x-l_x)u_y f(l_x,l_y)-l_x(u_y-l_y)f(u_x,u_y)+(l_x u_y-u_x l_y)f(l_x,u_y)}{(u_x-l_x)(u_y-l_y)} \quad .
\end{aligned}$$

Otherwise,

$$\mathrm{vex}_{[l,u]}[f](x,y) = \begin{cases} \alpha_1 x + \beta_1 y + \gamma_1, & \text{if } y_0 \le \frac{l_y-u_y}{u_x-l_x}(x_0-l_x)+u_y,\\ \alpha_2 x + \beta_2 y + \gamma_2, & \text{if } y_0 > \frac{l_y-u_y}{u_x-l_x}(x_0-l_x)+u_y, \end{cases}$$

where

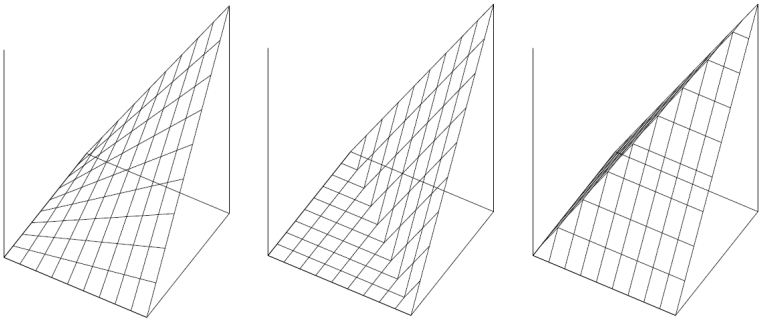
$$\begin{aligned}
\alpha_1 &= \tfrac{f(u_x,l_y)-f(l_x,l_y)}{u_x-l_x}, \quad \beta_1 = \tfrac{f(l_x,u_y)-f(l_x,l_y)}{u_y-l_y},\\
\gamma_1 &= \tfrac{(u_x u_y-l_x l_y)f(l_x,l_y)-l_x(u_y-l_y)f(u_x,l_y)-(u_x-l_x)l_y f(l_x,u_y)}{(u_x-l_x)(u_y-l_y)} \quad ,\\
\alpha_2 &= \tfrac{f(u_x,u_y)-f(l_x,u_y)}{u_x-l_x}, \quad \beta_2 = \tfrac{f(u_x,u_y)-f(u_x,l_y)}{u_y-l_y},\\
\gamma_2 &= \tfrac{(l_x l_y-u_x u_y)f(u_x,u_y)+u_x(u_y-l_y)f(l_x,u_y)+(u_x-l_x)u_y f(u_x,l_y)}{(u_x-l_x)(u_y-l_y)} \quad .
\end{aligned}$$

Example 3.14 (Bilinear functions [McC76]). Consider the bilinear term $f(x,y) = xy$ restricted to the box $[l,u] \subseteq \mathbf{R}^2$. The test $f(l_x,l_y) + f(u_x,u_y) \le f(l_x,u_y) + f(u_x,l_y)$ in Proposition 3.13 is equivalent to $l_x l_y + u_x u_y \le l_x u_y + u_x l_y$ and thus to $u_y(u_x - l_x) \le l_y(u_x - l_x)$, which is false for full dimensional boxes. The envelopes, displayed in Figure 3.3, are given by

$$\begin{aligned}
\mathrm{vex}_{[l,u]}[xy](x,y) &= \max\left\{ l_y x + l_x y - l_x l_y \ ,\ u_y x + u_x y - u_x u_y \right\},\\
\mathrm{cave}_{[l,u]}[xy](x,y) &= \min\left\{ u_y x + l_x y - l_x u_y \ ,\ l_y x + u_x y - u_x l_y \right\}.
\end{aligned}$$

◇

Meyer and Floudas [MF05] generalized the two dimensional criterion to three dimensional boxes. Up to symmetry there are 6 triangulation types of a box in dimension three. For each pair of vertices v^i, v^j, $i \ne j$, of the box which are not adjacent it is to check if the connecting line $\lambda f(v^i) + (1-\lambda) f(v^j)$ is *non-dominated* by another line or triangle at the



(a) $f(x, y) = xy$. (b) Convex envelope. (c) Concave envelope.

Figure 3.3.: Bilinear functions and their envelopes taken from [Lin05].

intersection of the two objects, i.e., it is to check whether

$$\lambda f(v^i) + (1-\lambda) f(v^j) \leq \mu_1 f(v^{k_1}) + \mu_2 f(v^{k_2}) + (1-\mu_1-\mu_2) f(v^{k_3}),$$

where $\lambda v^i + (1-\lambda)v^j = \mu_1 v^{k_1} + \mu_2 v^{k_2} + (1-\mu_1-\mu_2)v^{k_3}$, $\lambda, \mu_1, \mu_2 \geq 0$, and v^{k_l}, $l = 1, 2, 3$, are distinct from v^i and v^j. This leads to a set of subsets of non-dominated vertices of V. For instance, in Subfigure 3.2 (a) the set of non-dominated subsets is given by $\{\{v^1, v^3\}\}$ while it is $\{\{v^2, v^4\}\}$ in Subfigure 3.2 (b). The non-dominated subsets are then used to determine the triangulation type which leads to the convex envelope. See [MF05] for further details.

Example 3.15 (Trilinear functions [MF03, MF04]). Consider the trilinear product term $f(x, y, z) = xyz$ restricted to the nonnegative box $[l, u] \subseteq \mathbf{R}^3_{\geq 0}$. The appropriate triangulation can be determined by mapping the pairs $\{l_x, u_x\}$, $\{l_y, u_y\}$, and $\{l_z, u_z\}$ onto $\{l_1, u_1\}$, $\{l_2, u_2\}$, and $\{l_3, u_3\}$ in such a way that the following relations hold:

$$u_1 l_2 l_3 + l_1 u_2 u_3 \leq l_1 u_2 l_3 + u_1 l_2 u_3, \quad u_1 l_2 l_3 + l_1 u_2 u_3 \leq u_1 u_2 l_3 + l_1 l_2 u_3.$$

The convex envelope of $x_1x_2x_3$ is the maximum of the following equalities:

$$\begin{aligned}
\omega_1 =& l_2l_3x_1 + l_1l_3x_2 + l_1l_2x_3 - 2l_1l_2l_3,\\
\omega_2 =& u_2u_3x_1 + u_1u_3x_2 + u_1u_2x_3 - 2u_1u_2u_3,\\
\omega_3 =& l_2u_3x_1 + l_1u_3x_2 + u_1l_2x_3 - l_1l_2u_3 - u_1l_2u_3,\\
\omega_4 =& u_2l_3x_1 + u_1l_3x_2 + l_1u_2x_3 - u_1u_2l_3 - l_1u_2l_3,\\
\omega_5 =& \frac{\theta_1}{u_1 - l_1}x_1 + u_1l_3x_2 + u_1l_2x_3 + \left(\frac{\theta_1 l_1}{u_1 - l_1} - u_1u_2l_3 - u_1l_2u_3 + l_1u_2u_3\right),\\
& \text{where } \theta_1 = u_1u_2l_3 - l_1u_2u_3 - u_1l_2l_3 + u_1l_2u_3,\\
\omega_6 =& \frac{\theta_2}{l_1 - u_1}x_1 + l_1u_3x_2 + l_1u_2x_3 + \left(\frac{\theta_2 u_1}{l_1 - u_1} - l_1l_2u_3 - l_1u_2l_3 + u_1l_2l_3\right),\\
& \text{where } \theta_2 = l_1l_2u_3 - u_1l_2l_3 - l_1u_2u_3 + l_1u_2l_3.
\end{aligned}$$

The concave envelope is the minimum of the equalities given by

$$\begin{aligned}
\Omega_1 =& l_2l_3x_1 + u_1l_3x_2 + u_1u_2x_3 - u_1u_2l_3 - u_1l_2l_3,\\
\Omega_2 =& u_2l_3x_1 + l_1l_3x_2 + u_1u_2x_3 - u_1u_2l_3 - l_1u_2l_3,\\
\Omega_3 =& l_2l_3x_1 + u_1u_3x_2 + u_1l_2x_3 - u_1l_2u_3 - u_1l_2l_3,\\
\Omega_4 =& u_2u_3x_1 + l_1l_3x_2 + l_1u_2x_3 - l_1u_2u_3 - l_1u_2l_3,\\
\Omega_5 =& l_2u_3x_1 + u_1u_3x_2 + l_1l_2x_3 - u_1l_2u_3 - l_1l_2u_3,\\
\Omega_6 =& u_2u_3x_1 + l_1u_3x_2 + l_1l_2x_3 - l_1u_2u_3 - l_1l_2u_3.
\end{aligned}$$

◇

For larger dimensions a constructive approach similar to the one for dimension three is, in principle, possible. One obstacle is the a priori knowledge of all possible triangulations and the analysis of their properties. For instance, already in dimension four we have an explosion in the number of possible triangulations. The 4-cube exhibits 92,487,256 triangulations which can be partitioned into 247,451 symmetry classes [Pou13, HSYY08]. These huge numbers show that even the analysis of the triangulation classes is expensive and its implementation would be tedious.

Tawarmalani et al. [TRX12] consider component-wise concave functions f, whose restriction to the vertices of the box is *submodular*, i.e.,

$$f(v \wedge v') + f(v \vee v') \leq f(v) + f(v')$$

for all vertices v and v', where $v \wedge v'$ and $v \vee v'$ denote the component-wise minimum and maximum of v and v', respectively (cf. [KS12a]). For this subclass of component-wise concave functions the appropriate triangulation for the convex envelope is given by Kuhn's triangulation (cf. [TRX12]) and the convex envelope is known for any dimension. Note that in dimension four Kuhn's triangulation is only one of the 247,451 symmetry classes of possible triangulations. In Chapter 4 we return to component-wise concave functions and provide an extended formulation for their convex envelopes in arbitrary dimensions.

3.1.2. Indefinite and (n-1)-Convex Functions

The first work concerning functions with nonpolyhedral convex envelopes was accomplished by Tawarmalani and Sahinidis [TS02a, TS01]. They use disjunctive programming techniques to derive convex formulations for the epigraph of functions $f(x, y)$ which are component-wise concave in $x \in \mathbf{R}$ and convex in $y \in \mathbf{R}^n$. In particular, they derive the convex envelope of fractional terms x/y restricted to a box from the positive orthant. This function is not convex but component-wise convex in x and y, and it thus also belongs to the class of (n-1)-convex and indefinite functions investigated by Jach et al. [JMW08] and summarized in this subsection.

Definition 3.16 ([JMW08])**.** Let $f : \mathbf{R}^n \to \mathbf{R}$ be a twice differentiable function restricted to a convex domain $D \subseteq \mathbf{R}^n$.

- The function f is said to be *(n-1)-convex* over $\mathbf{R}^n$ if and only if for all $i \in \{1, \ldots, n\}$ the function $f|_{x_i = \bar{x}_i} : \mathbf{R}^{n-1} \to \mathbf{R}$ is convex for each fixed value $\bar{x}_i$.

- The function f is called *indefinite* (over D) if and only if for each $x \in D$ the Hessian $H_f(x)$ is indefinite.

An example of indefinite (n-1)-convex functions is $f(x, y) = x^2y^2$ over the box $[1, 2]^2$, whose Hessian exhibits the negative determinant $-12x^2y^2$. For indefinite (n-1)-convex functions Problem (VEX) simplifies considerably due to a geometrical property of their concave directions.

Definition 3.17 ([JMW08])**.** Let $f : \mathbf{R}^n \to \mathbf{R}$ be a twice differentiable function and $H_f(x)$ the Hessian matrix of f. The set of all concave directions of f at a point $x \in D$ is denoted by $\gamma_f(x) := \{y \in \mathbf{R}^n \mid y^\intercal H_f(x) y < 0\}$.

In terms of Observation 3.5 concave directions correspond to segments s over which f is concave such that all x in the relative interior of s can be excluded from the possible generating set. Jach et al. [JMW08] show that the set of concave directions of indefinite (n-1)-convex functions are always contained in one pair of opposite orthants.

Lemma 3.18 ([JMW08]). *Let $f : D \to \mathbf{R}$, $D = [l, u] \subseteq \mathbf{R}^n$, be a twice differentiable function, and let the collection $\{O_1, \ldots, O_{2^n}\}$ be the system of open orthants of the space $\mathbf{R}^n$. Then, the function f is (n-1)-convex and indefinite if and only if $\gamma_f(x)$ is nonempty for each $x \in D$ and there exists an index $i \in \{1, \ldots, 2^n\}$ such that*

$$\forall x \in D : \qquad \gamma_f(x) \quad \subseteq \quad O_i \cup (-O_i).$$

This result can be used to bound the number of points in (VEX) from above by two and to shrink the possible set of points contained in the generating set $G_D^{\text{vex}}[f]$ to the boundary of the box D.

Theorem 3.19 ([JMW08]). *Let $f : \mathbf{R}^n \to \mathbf{R}$ be an (n-1)-convex and indefinite function over $D := [l, u] \subseteq \mathbf{R}^n$, and denote the boundary of D by B. Then,*

$$\begin{aligned}\text{vex}_D[f](x) = \text{vex}_B[f](x) = \min\{&(1-\lambda)f(x^1) + \lambda f(x^2) \mid x^i \in B, i = 1, 2, \\ &(1-\lambda)x^1 + \lambda x^2 = x, 0 \le \lambda \le 1\}.\end{aligned}$$

In fact, this implies that the convex envelope of indefinite (n-1)-convex functions is the union of segments. More precisely, let $\bar{x}^1$ and $\bar{x}^2$ be the points used in Theorem 3.19 for a given point $\bar{x}$. Then, the convex envelope is affine along the segment connecting $\bar{x}^1$ and $\bar{x}^2$. See [JMW08] for further details.

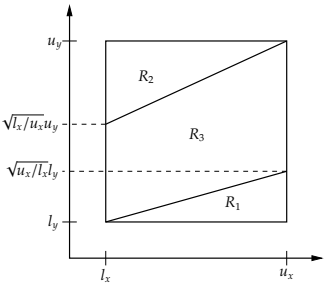
The theory of indefinite (n-1)-convex functions is illustrated for fractional terms x/y in Example 3.20 and for bivariate quadratic functions in Example 3.21.

Example 3.20 (Fractional terms [TS01, TS02a, JMW08]). Let $f(x, y) = x/y$ be restricted to a box $[l, u] := [l_x, u_x] \times [l_y, u_y] \subseteq \mathbf{R}_{\ge 0} \times \mathbf{R}_{>0}$. Its Hessian reads

$$H_f(x, y) = \begin{pmatrix} 0 & -\frac{1}{y^2} \\ -\frac{1}{y^2} & \frac{2x}{y^3} \end{pmatrix}.$$

The concave directions corresponding to f have to fulfill $\xi^\intercal H_f(x,y)\xi \leq 0$ which is equivalent to $\frac{2\xi_2(\xi_2 x - \xi_1 y)}{y^3} \leq 0$ and hence to $\xi_2(\xi_2 x - \xi_1 y) \leq 0$. For instance, if $\xi_2 \geq 0$, then ξ_1 needs to satisfy $\xi_2 x - \xi_1 y \leq 0$ which is equivalent to $\xi_2 \frac{x}{y} \leq \xi_1$ and thus, $(\xi_1, \xi_2) \in \mathbf{R}^2_{\geq 0}$. The set of concave directions $\gamma_f(x)$ is contained in $(\mathbf{R}^2_{\leq 0} \cup \mathbf{R}^2_{\geq 0})$ for all $x \in [l,u]$, which corresponds to Lemma 3.18.

Lemma 3.18 and Theorem 3.19 applied to x/y reveal three regions with different expressions for the convex envelope indicated by Figure 3.4 (a). This subdivision reflects the orientation of the concave directions. Note that the region R_3 vanishes if $\sqrt{u_x/l_x}\, l_y \geq u_y$ and $\sqrt{l_x/u_x}\, u_y \leq l_y$.



(a) Fractional term x/y.

Figure 3.4.: Fractional term x/y: Subdivision of the domain into three regions w.r.t. different expressions of the convex envelope.

The description of $\mathrm{vex}_{[l,u]}[x/y](x,y)$ over the regions R_1, R_2, and R_3 is given by:

$$R_1 : \frac{u_x - x}{u_x - l_x} \cdot \frac{l_x}{l_y} + \frac{x - l_x}{u_x - l_x} \cdot \frac{u_x}{\frac{y - l_y}{x - l_x}(u_x - x) + y},$$

$$R_2 : \frac{u_x - x}{u_x - l_x} \cdot \frac{l_x}{\frac{u_y - y}{u_x - x}(l_x - x) + y} + \frac{x - l_x}{u_x - l_x} \cdot \frac{u_x}{u_y},$$

$$R_3 : \frac{u_x - x}{u_x - l_x} \cdot \frac{l_x}{\frac{y\sqrt{l_x}(u_x - l_x)}{(u_x - x)\sqrt{l_x} + (x - l_x)\sqrt{u_x}}} + \frac{x - l_x}{u_x - l_x} \cdot \frac{u_x}{\frac{y\sqrt{u_x}(u_x - l_x)}{(u_x - x)\sqrt{l_x} + (x - l_x)\sqrt{u_x}}} = \frac{\left(x + \sqrt{l_x u_x}\right)^2}{y\left(\sqrt{l_x} \sqrt{u_x}\right)^2}.$$

$\diamond$

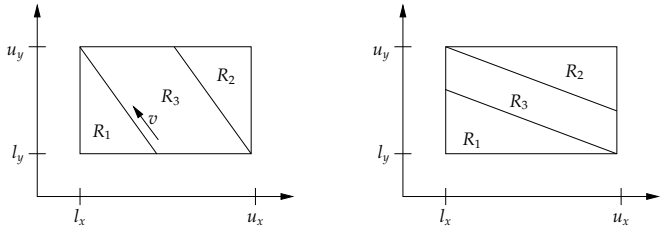
Example 3.21 (Bivariate quadratic functions [JMW08]). Let $f := a_1x + a_2y + 2a_{12}xy + a_{11}x^2 + a_{22}y^2 + c$ be a bivariate quadratic function restricted to a box $[l, u] \subseteq \mathbf{R}^2$ with $a_{12}, a_{11}, a_{22} \neq 0$. An analysis of the Hessian H_f leads to three different cases with respect to the convexity of f:

Case 1: H_f is positive or negative semidefinite. Then, f is convex or concave and the envelopes are straight forward to derive.

Case 2: H_f is indefinite and $a_{11}a_{22} \leq 0$. Assume $a_{11} > 0 \leq a_{22}$, i.e., f is convex in x and concave in y. Otherwise, consider $-f$. The convex envelope reads

$$\begin{aligned} \operatorname{vex}_{[l,u]}[f](x,y) &= \tfrac{u_y - y}{u_y - l_y}\, f(w_1, l_y) + (1 - \tfrac{u_y - y}{u_y - l_y})\, f(w_2, u_y), \quad \text{where} \\ w_1 &= \min\left\{ \tfrac{a_{12}}{a_{11}}(1 - \tfrac{u_y - y}{u_y - l_y})(u_y - l_y) + x, u_x, \tfrac{x - l_x}{y - u_y}(l_y - y) + x \right\}, \\ w_2 &= \max\left\{ \tfrac{a_{12}}{a_{11}} \tfrac{u_y - y}{u_y - l_y}(l_y - u_y) + x, \tfrac{x - u_x}{y - l_y}(u_y - y) + x, l_x \right\}. \end{aligned} \tag{3.3}$$

Case 3: In the last case we consider f as 1-convex function with an indefinite Hessian. Assume $a_{11}, a_{22} > 0$ and that the eigenvector to the negative eigenvalue has positive and negative entries such that the concave directions of f are contained in the pair of orthants $(\mathbf{R}_{\leq 0} \cup \mathbf{R}_{\geq 0}) \cup (\mathbf{R}_{\geq 0} \cup \mathbf{R}_{\leq 0})$. Otherwise, consider $f(x, l_y + u_y - y)$ which satisfies the later property. For the given orientation of the concave directions two subdivisions of the box are possible w.r.t. the description of the convex envelope (see Figure 3.5). If it holds that



(a) Subdiv. 1 ($v = (-\sqrt{a_{22}}, \sqrt{a_{11}})$). (b) Subdivision 2.

Figure 3.5.: Bivariate quadratic function: Subdivision of the domain into three regions with respect to different expressions of the convex envelope.

$$f(l_x, u_y) + \frac{\partial f}{\partial y}(l_x, u_y)(l_y - u_y) \geq f(u_x, l_y) + \frac{\partial f}{\partial x}(u_x, l_y)(l_x - u_x),$$

the subdivision in Figure 3.5 (a) yields the convex envelope. The formula for the convex envelope in region R_3 is given by Equation (3.3) while the expression for R_1 and R_2 are

$$\begin{aligned} \operatorname{vex}_{[l,u]}[f](x,y)|_{(x,y)\in R_1} &= \lambda f(\omega_1, l_y) \quad +(1-\lambda) f(l_x, \omega_2), \\ \operatorname{vex}_{[l,u]}[f](x,y)|_{(x,y)\in R_2} &= \mu f(\omega_3, u_y) \quad +(1-\mu) f(u_x, \omega_4), \end{aligned}$$

where $\lambda = \frac{x-l_x}{\omega_1 - l_x}$, $\mu = \frac{x-u_x}{\omega_3 - u_x}$, and

$$\begin{aligned} \omega_1 &= -\sqrt{\tfrac{a_{22}}{a_{11}}}(l_y - y) + x, & \omega_2 &= -\sqrt{\tfrac{a_{11}}{a_{22}}}(l_x - x) + y, \\ \omega_3 &= -\sqrt{\tfrac{a_{22}}{a_{11}}}(u_y - y) + x, & \omega_4 &= -\sqrt{\tfrac{a_{11}}{a_{22}}}(u_x - x) + y. \end{aligned}$$

Region R_3 may vanish depending on the box size and the vector v in Figure 3.5 (a). ◇

Remark 3.22. Note that the convex envelopes for bivariate fractional and bivariate quadratic functions are not given by one algebraic expression which is valid over the entire box but by three expressions, each of which is valid over a subdomain of the box (cf. Figures 3.4 and 3.5). This subdivision of the box into three regions w.r.t. the expressions of the convex envelopes is the general case for bivariate, indefinite (n-1)-convex functions and follows from the orientation of the concave directions (cf. [JMW08]).

3.1.3. Products of Convex and Component-Wise Concave Functions

Recently, Khajavirad and Sahinidis [KS12b, KS12a] deduced the convex envelope of functions $\phi(x, y) = f(x)g(y)$, where $f(x)$ is a univariate convex function and $g(y)$ is a multivariate component-wise concave function over boxes $[l, u] := [l_x, u_x] \times [l_y, u_y] \subseteq \mathbf{R} \times \mathbf{R}^n$. The authors argue that Problem (VEX) is equivalent to a convex problem in this case. For special classes of functions f closed-form expressions for the convex envelope are derived.

The generating set of $\operatorname{vex}_{[l,u]}[\phi]$ can be reduced to $G^{\text{vex}}_{[l,u]}[f] \subseteq \{(x, v) \mid x \in [l_x, u_x], v \in V\}$, where $V := \operatorname{vert}([l_y, u_y])$, because ϕ is component-wise concave in y. Problem (VEX) is further simplified by substituting $z^i = \lambda_i x^i$

in order to obtain the following *convex* optimization problem

$$\begin{aligned} \min_{z^v,\lambda_v} \quad & \sum_{v\in V} \lambda_v f(z^v/\lambda_v) g(v) \\ \text{s. t.} \quad & \sum_{v\in V} \lambda_v v = y, \quad \sum_{v\in V} z^v = x, \quad \sum_{v\in V} \lambda_v = 1, \\ & \lambda_v l_x \le z^v \le \lambda_v u_x, \quad \lambda_v \ge 0, \quad \forall v \in V. \end{aligned} \tag{CX1}$$

The term $\lambda_v f(z^v/\lambda_v)$ is the perspective function of the convex function $f(z^v)$ and thus convex itself (see [HUL01]). In order to solve (CX1) explicitly, the solution of Problem (VEX) for the component-wise concave function $g(y)$ is exploited. Problem (VEX) for $g(y)$ reads

$$\begin{aligned} \min_{\lambda_v} \quad & \sum_{v\in V} \lambda_v g(v) \\ \text{s. t.} \quad & \sum_{v\in V} \lambda_v v = y, \quad \sum_{v\in V} \lambda_v = 1, \\ & \lambda_v \ge 0, \quad \forall v \in V. \end{aligned} \tag{CCV}$$

Khajavirad and Sahinidis prove that for special classes of functions $f(x)$ and $g(y)$ the optimal multipliers λ_v in (CX1) are independent of x and coincide with the optimal multipliers in (CCV). Thus, (CX1) is reduced to the computation of the multipliers λ_v in (CCV) which is equivalent to the determination of the convex envelope of the component-wise concave function $g(y)$. For arbitrary dimensions the convex envelope of a component-wise concave function $g(y)$ is only known for functions whose restriction to the vertices of the box is submodular [TRX12] (see Section 3.1.1). In this case the multipliers λ_v, $v \in V$, are available which allows to compute the convex envelope of $\phi(x, y) = f(x)g(y)$.

Theorem 3.23 (Theorems 1 and 3 in [KS12b], Theorem 1 in [KS12a])**.** *Let $\phi(x, y) = f(x)g(y)$ be restricted to the box $[l, u] := [l_x, u_x] \times [l_y, u_y] \subseteq \mathbf{R} \times \mathbf{R}^n$, where*

- *$f(x)$ is a nonnegative convex function of one of the two forms (i) $f(x) = x^a$, $a \in \mathbf{R} \setminus [0, 1]$ or (ii) $f(x) = a^x, a > 0$,*
- *$g(y)$ is a component-wise concave function such that its restriction to the vertices is submodular and has the same monotonicity in every argument, and*

- *$f(x)$ is monotone or $g(y)$ is nonnegative.*

Then, the values of the optimal multipliers in (CX1) *are independent of x and correspond to the Lovász extension of $g(y)$ restricted to the vertices of the box. If $g(y)$ is nonnegative, the univariate variable x in $f(x)$ can be replaced by $c^\top x + d$, where x is multivariate.*

Khajavirad and Sahinidis remark that functions $g(y)$ which are not submodular or monotone may satisfy these assumptions after an affine transformation T of the variables. In this case the relation $\text{vex}_D[f(x)g(y)](x, y) = \text{vex}_D[f(x)g(T(y))](x, T(y))$ can be employed. Moreover, the assumption of $g(y)$ being a component-wise concave function can be relaxed to functions which exhibit a vertex polyhedral convex envelope over $[l_y, u_y]$.

Special attention is paid to univariate and bivariate functions $g(y)$ in order to relax some assumptions on f and g. In the univariate case the authors exploit that the domain is an interval so that the multipliers λ_v are unique. For bivariate functions Proposition 3.13 states the convex envelope of $g(y)$ which induces the multipliers λ_v. This is used for the next result.

Proposition 3.24 (Lemma 4, Propositions 3 and 5 in [KS12a] and Proposition 3 in [KS12a])**.** *Let f and g be defined as in Theorem 3.23 with $g : \mathbf{R}^2 \to \mathbf{R}$. Denote by $\hat{g}(y)$ the restriction of $g(y_1, y_2)$ to vertices of $[l_y, u_y]$. Then, the optimal multipliers in the description of $\text{vex}_{[l_y,u_y]}[g]$ are also optimal for the convex envelope of $\phi(x, y) = f(x)g(y)$ over $[l_x, u_x] \times [l_y, u_y]$ if one of the following conditions is satisfied for $\hat{g}(y)$:*

(i) It is submodular and nondecreasing (or nonincreasing) in both arguments.

(ii) It is supermodular, nondecreasing in y_1 and nonincreasing in y_2.

(iii) It is nonmonotone in at least one argument or is constant over any edge of $[l_y, u_y]$.

The discussion of convex envelopes is completed by functions whose convex envelope is described by pair-wise complementary convex combinations. See [Taw10]. An example for these functions is $f(x, y) := \exp(-xy)$ restricted to $[l, u] = [-1, 1] \times [-2, 0]$, i.e., functions which are nondecreasing and convex for $x = l_x$, and nonincreasing and convex for $x = u_x$.

We emphasize that this brief summary of convex envelopes covers the major part of available convex envelopes over box domains. For

many classes of functions the convex envelope is not known, e.g., xyz^2 or second-order isotherms given in Equation (3.1). Nevertheless, tight convex relaxations are essential for optimization issues. Therefore, we briefly name three alternative relaxation methods in order to conclude this section. First, Maranas and Floudas propose a method for twice continuously differentiable functions, where the nonconvex characteristics of the function over a box are overpowered by the addition of a nonpositive convex quadratic term [MF94, ADFN98, AF04a, AF04b]. Second, McCormick [McC76] suggests a bounding strategy for the compositions of functions $f := g(h(x))$, where $h : D \subseteq \mathbf{R}^n \to \mathbf{R}$ and $g : \mathbf{R} \to \mathbf{R}$ which is based on the convex envelope of the univariate function g. A third alternative is given by the *lifting technique* which we discuss in Section 3.2.2.

3.2. A Cut-Generation Algorithm for Bivariate Functions

In this section we present the results of a cut-generation algorithm for bivariate, twice continuously differentiable functions $f : [l, u] \subseteq \mathbf{R}^2 \to \mathbf{R}$, $(x, y) \mapsto f(x, y)$, with a *fixed convexity behavior*, i.e., the signs of the second partial derivatives w.r.t. each of the variables and the determinant of the Hessian are independent of a given point $(x, y) \in [l, u]$. For this, we elaborate and implement the results of Tawarmalani and Sahinidis [TS01], and Jach et al. [JMW08] (see Section 3.1.2). Their findings allow us to compute the value of the convex envelope at a given point numerically and then to construct supporting hyperplanes on the convex envelopes which are used to cut-off solutions of a current relaxation. The cut-generation algorithm is implemented in the open-source, mixed-integer nonlinear optimization solver SCIP [Ach07, Ach09] and it is available in its standard distribution from version 2.1 onwards. The results in this section are a summary of the technical report [BMV13].

In general, the convexity behavior of a function depends on the underlying domain. This makes it hard to determine whether a given function has a fixed convexity behavior. For bivariate quadratic functions $f(x, y) = a_{x,x}x^2 + a_{x,y}xy + a_{y,y}y^2 + b_x x + b_y y + c$ and monomial functions $f(x, y) = x^p y^q$, $p, q \in \mathbf{R}$, the convexity behavior can be checked easily. The Hessian of a bivariate quadratic function has only constant entries so that the function has the same convexity behavior at any point $(x, y) \in \mathbf{R}^2$. The Hessian of a bivariate monomial function can also have

nonconstant entries. Nevertheless, if we restrict such a function to nonnegative domains, the convexity behavior is fixed. In Table 3.1 we state the five fixed convexity behaviors needed to determine the convex envelope and the corresponding criteria for bivariate quadratic and monomial functions. The concave envelope can be derived analogously as $\operatorname{cave}_{[l_x,u_x]\times[l_y,u_y]}[f] = -\operatorname{vex}_{[l_x,u_x]\times[l_y,u_y]}[-f]$.

Convexity of f	$a_{x,x}x^2 + a_{x,y}xy + a_{y,y}y^2$ $+b_x x + b_y y + c$	$x^p y^p$, $(x,y) \in \mathbf{R}^2_{\geq 0}$
1. convex	$a_{x,x} \geq 0, a_{y,y} \geq 0,$ $a_{x,x}a_{y,y} - a_{x,y}^2 \geq 0$	$p^2 - p \geq 0, q^2 - q \geq 0,$ $pq(1-p-q) \geq 0$
2. concave in x, y	$a_{x,x} \leq 0$, $a_{y,y} \leq 0$	$p^2 - p \leq 0$, $q^2 - q \leq 0$
3. strictly convex in x, concave in y	$a_{x,x} > 0$, $a_{y,y} \leq 0$	$p^2 - p > 0$, $q^2 - q \leq 0$
4. concave in x, strictly convex in y	$a_{x,x} \leq 0$, $a_{y,y} > 0$	$p^2 - p < 0$, $q^2 - q \leq 0$
5. not convex, but strictly convex in x, y	$a_{x,x} > 0, a_{y,y} > 0,$ $a_{x,x}a_{y,y} - a_{x,y}^2 < 0$	$p^2 - p \geq 0, q^2 - q \geq 0,$ $pq(1-p-q) < 0$

Table 3.1.: Classes of fixed convexity behavior and criteria for bivariate quadratic and bivariate monomial functions. The latter functions are restricted to $[l,u] \subseteq \mathbf{R}^2_{\geq 0}$.

For functions with a convexity behavior corresponding to cases 1 and 2 in Table 3.1, explicit formulas for the convex envelopes are known. For general functions belonging to cases 3, 4, and 5, only structural results are available: Locatelli and Schoen [LS10, Loc10] provide a framework in which supporting hyperplanes on the convex envelopes can be computed directly. Their approach is based on the capability to solve a series of three-dimensional convex problems. In contrast to this, we apply the results of [TS01, JMW08] which do not directly yield supporting hyperplanes but the value of the convex envelope. However, this approach only requires solving one-dimensional convex problems corresponding to Problem (VEX) whose solutions can be used to construct supporting hyperplanes on the convex envelope. We implement the ideas of

[TS01, JMW08] in a way that we numerically solve the one-dimensional convex optimization problems and exploit the solutions to generate supporting hyperplanes in a separation algorithm.

The remainder of this section follows the structure of our cut-generation algorithm which consists of two subroutines. The first subroutine is based on the evaluation of the convex envelope of f and is discussed in Section 3.2.1. Let (x_0, y_0) be the solution of the current relaxation. If (x_0, y_0) is in the interior of the box $[l, u]$, we solve (VEX) at (x_0, y_0). The solution of this problem can be used to construct a *maximally touching, underestimating hyperplane*, i.e., a hyperplane which is not dominated by another underestimating hyperplane. If (x_0, y_0) is in the boundary of the box, the solution of (VEX) may only provide an underestimator which is valid over a facet of the box. In Section 3.2.2 we apply a lifting technique in the second subroutine to extend this locally valid underestimator to the entire box. The presented ideas are implemented in the constraint handler "cons_bivariate" in SCIP. In Section 3.2.3 a computational case study illustrates the performance of the new constraint handler compared to state-of-the-art solvers.

3.2.1. Cuts from the Convex Envelope

Consider a bivariate function $f : [l, u] \subseteq \mathbf{R}^2 \to \mathbf{R}$, $(x, y) \mapsto f(x, y)$, with a fixed convexity behavior over $[l, u] \subseteq \mathbf{R}^2$ according to Table 3.1. We deal with each of the 5 convexity patterns separately to deduce maximally touching hyperplanes on the convex envelope of f. Assume $[l, u] := [l_x, u_x] \times [l_y, u_y] \subseteq \mathbf{R}^2$ with $l_x < u_x$ and $l_y < u_y$.

Case 1: $f(x, y)$ is convex

The convex envelope of a convex function is the function itself. Thus, the best possible linear underestimator of f at (x_0, y_0) is given by the tangent plane:

$$f(x, y) \quad \geq \quad \nabla f(x_0, y_0)^\top \Big((x, y) - (x_0, y_0) \Big) + f(x_0, y_0).$$

Case 2: $f(x, y)$ is concave in x and y

According to [McC76, Tar03, KS12a], the convex envelope of a bivariate component-wise concave function f is given by Proposition 3.13.

For the remainder of Subsection 3.2.1 we assume that the given point (x_0, y_0) is in the *interior* of $[l, u]$. The cases where (x_0, y_0) is in the boundary of $[l, u]$ are discussed in Subsection 3.2.2.

Case 3: $f(x, y)$ is strictly convex in x and concave in y

We can infer from Observations 3.5 and 3.6 that $G^{\text{vex}}_{[l,u]}[f] = [l_x, u_x] \times \{l_y, u_y\}$. The minimization problem (VEX) corresponding to the evaluation of the convex envelope at a given point (x_0, y_0) can be thus simplified to (cf. [TS01, JMW08]):

$$\begin{aligned} \text{vex}_{[l,u]}[f](x_0, y_0) \quad = \quad & \min \quad tf(r, l_y) + (1-t)f(s, u_y) \\ & \text{s.t.} \quad \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = t \begin{pmatrix} r \\ l_y \end{pmatrix} + (1-t) \begin{pmatrix} s \\ u_y \end{pmatrix}, \\ & \quad\quad 0 \le t \le 1, \quad r, s \in [l_x, u_x]. \end{aligned} \tag{3.4}$$

As $l_x < x_0 < u_x$ and $l_y < y_0 < u_y$, we can use the identities $r(s) = \frac{l_y - u_y}{y_0 - u_y} x_0 - \frac{l_y - y_0}{y_0 - u_y} s$ and $t = \frac{y_0 - u_y}{l_y - u_y}$ to rewrite Problem (3.4) into the following *univariate convex* problem:

$$\min v_{\text{red}}(s) \ \text{ s.t. } \max\left\{l_x, \tfrac{y_0-u_y}{l_y-y_0}\left[\tfrac{l_y-u_y}{y_0-u_y}x_0 - u_x\right]\right\} \le s \le \min\left\{\tfrac{y_0-u_y}{l_y-y_0}\left[\tfrac{l_y-u_y}{y_0-u_y}x_0 - l_x\right], u_x\right\}, \tag{3.5}$$

where $v_{\text{red}}(s)$ reads

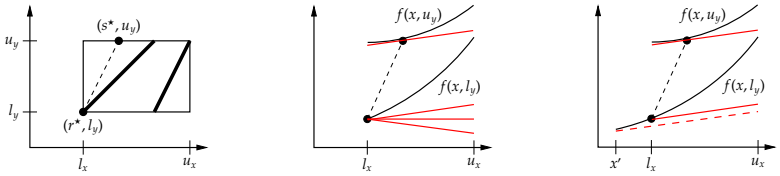
$$v_{\text{red}}(s) := \tfrac{y_0-u_y}{l_y-u_y} f\Big(\tfrac{l_y-u_y}{y_0-u_y}x_0 - \tfrac{l_y-y_0}{y_0-u_y}s, l_y\Big) + \tfrac{l_y-y_0}{l_y-u_y} f(s, u_y),$$

and the constraints in Problem (3.5) result from $l_x \le s \le u_x$ and $l_x \le r(s) \le u_x$. To solve the convex Problem (3.5), we use a Newton's method to determine a root of the first derivative of $v_{\text{red}}(s)$. If the root is not contained in the feasible region of Problem (3.5), the minimum is attained at the lower or upper bound of s. Let s^* denote an optimal solution of the reduced Problem (3.5). Then, the point (s^*, r^*, t^*), with $r^* = r(s^*) = \frac{l_y-u_y}{y_0-u_y}x_0 - \frac{l_y-y_0}{y_0-u_y}s^*$ and $t^* = \frac{y_0-u_y}{l_y-u_y}$, is an optimal solution of Problem (3.4), i.e., $\text{vex}_{[l,u]}[f](x_0, y_0) = v_{\text{red}}(s^*)$.

It remains to compute a maximally touching hyperplane $h(x, y)$ on the convex envelope. By construction, $\text{vex}_{[l,u]}[f](x, y)$ is linear over the segment connecting (r^*, l_y) and (s^*, u_y) which contains the point (x_0, y_0), i.e., the maximally touching hyperplane and $\text{vex}_{[l,u]}[f](x, y)$ coincide along the

segment. A maximally touching hyperplane on the graph of $\text{vex}_{[l,u]}[f]$ at (x_0, y_0) is therefore defined by the points $p_1 = (r^*, l_y, f(r^*, l_y))$, $p_2 = (s^*, u_y, f(s^*, u_y))$, and a direction vector q. To determine q, we consider the restriction of $f(x, y)$ and $h(x, y)$ to the facets $\bar{y} \in \{l_y, u_y\}$, where f is convex. Thus, $h(x, \bar{y})$ needs to underestimate the tangents on f at (r^*, l_y) and (s^*, u_y) over $[l_x, u_x]$. If $r^*, s^* \in (l_x, u_x)$, the only underestimating and touching hyperplane is the tangent on the convex function $f(x, \bar{y})$. This implies $q = (1, 0, \frac{\partial f}{\partial x}(r^*, l_y)) = (1, 0, \frac{\partial f}{\partial x}(s^*, u_y))$ (see [JMW08]).

If, for instance, $r^* = l_x$ and $l_x < s^* < u_x$ (cf. Figure 3.6 (a)), there are several underestimating and touching hyperplanes at (r^*, l_y) along the x-direction while the touching hyperplane at (s^*, u_y) is unique and equivalent to the tangent on f at this point (cf. Figure 3.6 (b)). Figure 3.6 (c) indicates that a parallel shift of the tangent at (s^*, u_y) to (r^*, l_y) leads also to a valid underestimator along $y = l_y$. Thus, $q = (1, 0, \frac{\partial f}{\partial x}(s^*, u_y))$.



(a) Subdivision (bold). (b) Underest. (red). (c) Shifted underest.

Figure 3.6.: Figure (a) depicts the subdivision of the box if $s^* = l_x$. Figures (b) and (c) show different valid underestimators (red) for a function (black).

In general, the direction vector is given by $q = (1, 0, \frac{\partial f}{\partial x}(\bar{x}, \bar{y}))$, where the point $(\bar{x}, \bar{y}) \in \{(r^*, l_y), (s^*, u_y)\}$ has to be chosen as follows. If $s^* \in (l_x, u_x)$, set $(\bar{x}, \bar{y}) = (s^*, u_y)$. If $s^* \in \{l_x, u_x\}$ and $r^* \in (l_x, u_x)$, set $(\bar{x}, \bar{y}) = (r^*, l_y)$. Otherwise, both points $(\bar{x}^1, \bar{y}^1) = (r^*, l_y)$ and $(\bar{x}^2, \bar{y}^2) = (s^*, u_y)$ yield valid inequalities.

Case 4: $f(x, y)$ is concave in x and strictly convex in y

Switch the variables and apply the procedure in the previous case.

Case 5: $f(x, y)$ is not convex, but strictly convex in x and y

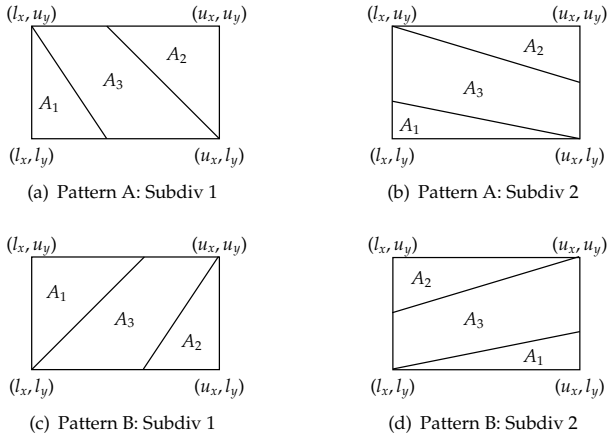
With Theorem 3.19 the value of the convex envelope of f over $[l, u]$ at (x_0, y_0) is given by

$$\begin{aligned} \min \quad & tf(x_1, y_1) + (1-t)f(x_2, y_2) \\ \text{s. t.} \quad & \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = t \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + (1-t) \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}, \\ & 0 \leq t \leq 1, \quad (x_1, y_1), (x_2, y_2) \in B, \end{aligned} \tag{3.6}$$

where B denotes the boundary of the box $[l, u]$, i.e., $x_1 \in \{l_x, u_x\}$ or $y_1 \in \{l_y, u_y\}$, and $x_2 \in \{l_x, u_x\}$ or $y_2 \in \{l_y, u_y\}$. A case distinction can be used to simplify Problem (3.6) which assigns (x_1, y_1) and (x_2, y_2) to different facets of the box. We obtain six simplified optimization problems because two assignments to parallel facets and four assignments to orthogonal facets of the box have to be considered. The minimum of all 6 cases yields the value of the convex envelope. According to Lemma 3.18 this case distinction can be avoided because the concave directions of indefinite (n-1)-convex functions are contained in a pair of orthants of $\mathbf{R}^2$. To determine this pair for a given function, we can compute the eigenvector to the negative eigenvalue of the Hessian $\mathcal{H}_f(\bar{x}, \bar{y})$ of f at the midpoint of the box, for example. If the eigenvector has entries with different signs, the concave directions of f at any point in $[l, u]$ are contained in the union $(\mathbf{R}_{\geq 0} \times \mathbf{R}_{\leq 0}) \cup (\mathbf{R}_{\leq 0} \times \mathbf{R}_{\geq 0})$ [**pattern A**]. Otherwise, the concave directions are contained in the union $(\mathbf{R}_{\geq 0} \times \mathbf{R}_{\geq 0}) \cup (\mathbf{R}_{\leq 0} \times \mathbf{R}_{\leq 0})$ [**pattern B**].

In Remark 3.22 we pointed out that the convex envelope of bivariate, indefinite (n-1)-convex functions is described by at most three expressions which correspond to a specific subdivision of the box. Each pattern of the concave directions leads to two possible structures for the subdivision of the box w.r.t. the description of the convex envelope as depicted in Figure 3.7.

We concentrate on pattern A in the following, as the structures of pattern B correspond to the ones of A if they are mirrored along a vertical line. Formally this can be described as follows. Define $\tilde{f}(x, y) := f(x, l_y + u_y - y)$ and note that $(l_y + u_y - y) \in [l_y, u_y]$ for all $y \in [l_y, u_y]$. The set of concave directions $\tilde{f}(x, y)$ matches pattern A. Then, using relations $f(x, y) = \tilde{f}(x, l_y + u_y - y)$ and $\operatorname{vex}_{[l,u]}[f](x, y) = \operatorname{vex}_{[l,u]}[\tilde{f}](x, l_y + u_y - y)$, the convex envelope of a function f belonging to pattern B can be determined



(a) Pattern A: Subdiv 1 (b) Pattern A: Subdiv 2

(c) Pattern B: Subdiv 1 (d) Pattern B: Subdiv 2

Figure 3.7.: Possible subdivisions of a box w.r.t. the description of the convex envelope.

by the arguments for pattern A.

Assuming pattern A the shape of the subdivision of the box w.r.t. the description of the convex envelope can be determined according to the next result. This further reduces the number of possible assignments of the endpoints (x_1, y_1) and (x_2, y_2) of the minimizing segment to the facets.

Lemma 3.25 (cf. Example 5 in [JMW08])**.** *The structure for the subdivision of the convex envelope corresponds to Figure 3.7 (a) if* $f(l_x, u_y) + (l_y - u_y)\frac{\partial f}{\partial y}(l_x, u_y) \geq f(u_x, l_y) + (l_x - u_x)\frac{\partial f}{\partial x}(u_x, l_y)$*. Otherwise, the structure follows Figure 3.7 (b).*

Subsequently, we discuss the convex envelope with a subdivision as in Figure 3.7 (a). The formulas corresponding to the subdivision in Figure 3.7 (b) are derived analogously by interchanging x and y.

Note that Lemma 3.25 only provides information about the general shape of the subdivision but not about the concrete shape of the domains A_1, A_2, and A_3 in Figure 3.7 (a). To determine a minimizing segment for a given point, we solve two auxiliary problems. The minimal value of the two problems is then equivalent to the value of the convex envelope. The first auxiliary problem corresponds to subdomain A_3, where

the endpoints of the possible minimizing segment are contained in the parallel facets given by $y = l_y$ and $y = u_y$. The second auxiliary problem corresponds to the subdomains A_1 and A_2 depending on the position of the point (x_0, y_0). If the point (x_0, y_0) is below the diagonal of the box connecting (l_x, u_y) and (u_x, l_y), the endpoints of the possible minimizing segment are contained in the orthogonal facets $x = l_x$ and $y = l_y$ (subdomain A_1). Otherwise, the endpoints of the possible minimizing segment are contained in the orthogonal facets $x = u_x$ and $y = u_y$ (subdomain A_2).

Auxiliary Problem 1: Parallel Facets In this case we have $(x_1, y_1) = (r, l_y)$ and $(x_2, y_2) = (s, u_y)$ in Problem (3.6) which then reduces to

$$\begin{aligned} \varrho(x_0, y_0) \quad := \quad & \min \quad tf(r, l_y) + (1-t)f(s, u_y) \\ & \text{s.t.} \quad \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = t\begin{pmatrix} r \\ l_y \end{pmatrix} + (1-t)\begin{pmatrix} s \\ u_y \end{pmatrix}, \\ & \qquad\quad 0 \le t \le 1, \quad r, s \in [l_x, u_x]. \end{aligned} \tag{3.7}$$

This subproblem is identical to the case considered in Subsection 3.2.1.

Auxiliary Problem 2: Orthogonal Facets If (x_0, y_0) is below the diagonal, i.e., $y_0 \le \frac{l_y - u_y}{u_x - l_x}(x_0 - l_x) + u_y$, we set $(x_1, y_1) = (l_x, r)$ and $(x_2, y_2) = (s, l_y)$ in Problem (3.6) and obtain

$$\begin{aligned} \omega_1(x_0, y_0) \quad := \quad & \min \, tf(l_x, r) + (1-t)f(s, l_y) \\ & \text{s.t.} \, \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = t\begin{pmatrix} l_x \\ r \end{pmatrix} + (1-t)\begin{pmatrix} s \\ l_y \end{pmatrix}, \\ & \qquad\quad 0 < t < 1, \; r \in [l_y, u_y], \; t \in [l_x, u_x]. \end{aligned} \tag{3.8}$$

Following [JMW08] the transformations $s(t) = (x_0 - l_x t)/(1-t)$ and $r(t) = (y_0 - (1-t)l_y)/t$ can be used to reformulate Problem (3.8) into the following *univariate convex* problem

$$\min \; tf\Big(l_x, \tfrac{y_0 - (1-t)l_y}{t}\Big) \; + \; (1-t)f\Big(\tfrac{x_0 - l_x t}{1-t}, l_y\Big) \quad \text{s.t.} \; t \in \Big[\tfrac{y_0 - l_y}{u_y - l_y}, \tfrac{u_x - x_0}{u_x - l_x}\Big], \tag{3.9}$$

where the constraint is induced by $l_x \le s(t) \le u_x$ and $l_y \le r(t) \le u_y$. Numerical methods can be applied to determine an optimal solution of Problem (3.9). Let $t^* \in (0, 1)$ be such an optimal solution. Then, the point

(t^*, s^*, r^*) with $s^* = s(t^*)$ and $r^* = r(t^*)$ is an optimal solution of Problem (3.8).

If the point (x_0, y_0) is above the diagonal, i.e., $y_0 > \frac{l_y - u_y}{u_x - l_x}(x_0 - l_x) + u_y$, we set $x_1 = u_x$ and $y_2 = u_y$ in Problem (3.6) and obtain the problem

$$\begin{aligned} \omega_2(x_0, y_0) \quad := \quad & \min\ tf(u_x, r) + (1-t)f(s, u_y) \\ & \text{s.t. } \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = t\begin{pmatrix} u_x \\ r \end{pmatrix} + (1-t)\begin{pmatrix} s \\ u_y \end{pmatrix}, \qquad (3.10) \\ & 0 < t < 1,\ r \in [l_y, u_y],\ s \in [l_x, u_x]. \end{aligned}$$

Problem (3.10) can be solved analogously to Problem (3.8).

Thus, the value of the convex envelope is the minimum of the optimal value $\varrho(x_0, y_0)$ of Problem (3.7) corresponding to the parallel case and of either $\omega_1(x_0, y_0)$ or $\omega_2(x_0, y_0)$ of Problems (3.8) and (3.10), respectively, corresponding to the orthogonal case:

$$\text{vex}_{[l,u]}[f](x_0, y_0) = \begin{cases} \min\{\varrho(x_0, y_0), \omega_1(x_0, y_0)\}, & \text{if } y_0 \leq \frac{l_y - u_y}{u_x - l_x}(x_0 - l_x) + u_y, \\ \min\{\varrho(x_0, y_0), \omega_2(x_0, y_0)\}, & \text{if } y_0 > \frac{l_y - u_y}{u_x - l_x}(x_0 - l_x) + u_y. \end{cases}$$

To construct supporting hyperplanes, we thus need to consider three cases:

(i) $\text{vex}_{[l,u]}[f](x_0, y_0) = \varrho(x_0, y_0)$: A minimizing segment is given by the optimal solution of Problem (3.7). The formulas for a linear underestimator can be derived as in Case 3, where f is convex in x and concave in y.

(ii) $y_0 \leq \frac{l_y - u_y}{u_x - l_x}(x_0 - l_x) + u_y$ and $\text{vex}_{[l,u]}[f](x_0, y_0) = \omega_1(x_0, y_0)$: Let (t^*, s^*, r^*) denote an optimal solution of Problem (3.8). A touching hyperplane on the graph of the convex envelope at the point (x_0, y_0) is given by the two points $p_1 = (l_x, r^*, f(l_x, r^*))$, $p_2 = (s^*, l_y, f(s^*, l_y))$, and a direction vector q. Similar to the arguments in case 4 for convex/concave functions, the vector q can be determined as follows. If $s^* \neq u_x$, then $q = (1, 0, \frac{\partial f}{\partial x}(s^*, l_y))$. If $s^* = u_x$ and $r^* \neq u_y$, then $q = (0, 1, \frac{\partial f}{\partial y}(l_x, r^*))$. If $s^* = u_x$ and $r^* = u_y$, then $q = (1, 0, \min\{\frac{\partial f}{\partial x}(s^*, l_y), \frac{\partial f}{\partial x}(l_x, r^*)\})$.

(iii) $y_0 > \frac{l_y - u_y}{u_x - l_x}(x_0 - l_x) + u_y$ and $\text{vex}_{[l,u]}[f](x_0, y_0) = \omega_2(x_0, y_0)$: This case can be handled analogously to (ii).

3.2.2. Cuts from the Lifting Technique

In the previous subsection we presented valid cuts for convex functions and functions being concave in each variable (cases 1 and 2 in Table 3.1). For cases 3, 4, and 5 we computed supporting hyperplanes for a given point (x_0, y_0) in the interior of the domain $[l, u]$. If (x_0, y_0) is in the boundary of $[l, u]$, the segment connecting the optimal solutions of Problems (3.4) and (3.6) is contained in a facet of $[l, u]$. Thus, the resulting underestimating hyperplane may only be valid over this facet. In this subsection we apply a *lifting technique* to extend such a *locally* valid underestimator to the entire box.

The concept of lifting techniques was introduced by Padberg [Pad75] to compute tight linear inequalities for linear zero-one problems. It was adopted in [GKH+06, GHJ+08b] to derive linear and convex underestimators for concrete examples of nonlinear functions over continuous domains. Some first general results in the field of continuous programs can be found in [RT10].

In our setting the key idea of the lifting procedure is the following [GHJ+08b]. Given a bivariate function $f : \mathbf{R}^2 \to \mathbf{R}$, $(x, y) \mapsto f(x, y)$, over a box $[l, u] := [l_x, u_x] \times [l_y, u_y] \subseteq \mathbf{R}^2$. We first fix one variable to one of its bounds, e.g., $x = l_x$. The univariate function $f(l_x, y)$ over $[l_y, u_y]$ is either convex or concave in our context so that its best underestimator $g : \mathbf{R} \to \mathbf{R}$ is a tangent or a secant, respectively. Our aim is to determine a best possible *lifting coefficient* $\mu \in \mathbf{R}$ such that

$$f(x, y) \quad \geq \quad \mu(x - l_x) + g(y) \qquad \text{for all } (x, y) \in [l, u].$$

This gives rise to the following nonlinear optimization problem

$$\mu \quad := \quad \inf\left\{ \tfrac{f(x,y)-g(y)}{x-l_x} \mid x \in (l_x, u_x],\ y \in [l_y, u_y] \right\}. \tag{3.11}$$

On the other hand, if we fix x to its upper bound u_x and assume that $h : \mathbf{R} \to \mathbf{R}$ is an underestimating function for $f(u_x, y)$ over $[l_y, u_y]$, we determine a best possible number $\tau \in \mathbf{R}$ with

$$f(x, y) \quad \geq \quad \tau(x - u_x) + h(y) \qquad \text{for all } (x, y) \in [l, u].$$

Using that $x - u_x \leq 0$ for $x \in [l_x, u_x]$, we end up with the following

optimization task

$$\tau \quad := \quad \sup\left\{\tfrac{f(x,y)-h(y)}{x-u_x} \mid x \in [l_x, u_x),\ y \in [l_y, u_y]\right\}. \tag{3.12}$$

In general, Problems (3.11) and (3.12) can be extremely difficult to solve. We exploit the specific structure of our bivariate functions to determine appropriate lifting coefficients.

To complete our cut-generation algorithm from the previous section, we have to investigate the lifting

1. from a facet over which the function is concave into a direction in which the function is convex (cases 3 and 4 in Table 3.1),

2. from a facet over which the function is convex into a direction in which the function is concave (cases 3 and 4 in Table 3.1), and

3. from a facet over which the function is convex into a direction in which the function is convex (case 5 in Table 3.1).

For this, we use elementary arguments that were also exploited in the papers [GKH+06, GHJ+08b].

Lifting from a facet over which the function is concave into a direction in which the function is convex

Let $f : [l, u] \to \mathbf{R}$ be a bivariate function that is convex in x and concave in y. Consider the point (x_0, y_0) with $x_0 \in \{l_x, u_x\}$ and $l_y \le y_0 \le u_y$. As f is concave in y, the best linear underestimator for $f(x_0, y)$ over $[l_y, u_y]$ is given by the secant $g : \mathbf{R} \to \mathbf{R}$ on the graph of $f(x_0, y)$ through the points $(x_0, l_y, f(x_0, l_y))$ and $(x_0, u_y, f(x_0, u_y))$, i.e.,

$$g(y) \quad := \quad \frac{f(x_0, u_y) - f(x_0, l_y)}{u_y - l_y}(y - l_y) + f(x_0, l_y).$$

Next, we determine the lifting coefficient μ according to Equation (3.11).

Case (a): $x_0 = l_x$. We will argue that

$$\mu = \tfrac{\partial f}{\partial x}(l_x, \bar{y}), \quad \text{where} \quad \bar{y} := \begin{cases} l_y, & \text{if } \tfrac{\partial f}{\partial x}(l_x, u_y) \ge \tfrac{\partial f}{\partial x}(l_x, l_y), \\ u_y, & \text{otherwise.} \end{cases}$$

The underestimator and the function coincide at $x = l_x$ and $y \in \{l_y, u_y\}$, so that $\mu \leq \frac{\partial f}{\partial x}(l_x, \bar{y})$ since we lift into a direction in which the function is convex. Along the line $y = \bar{y}$ the lifting coefficient $\frac{\partial f}{\partial x}(l_x, \bar{y})$ is best possible. The resulting linear underestimator is valid for f over $[l, u]$ because (i) it underestimates f along the lines $y = l_y$ and $y = u_y$ and (ii) it underestimates f along each segment from (x, l_y) to (x, u_y) for all $x \in [l_x, u_x]$ as it is linear in y while f is concave in y.

Case (b): $x_0 = u_x$. Analogously to case (a), the best lifting coefficient is given by

$$\tau = \tfrac{\partial f}{\partial x}(u_x, \bar{y}), \quad \text{where } \bar{y} := \begin{cases} l_y, & \text{if } \frac{\partial f}{\partial x}(u_x, u_y) \leq \frac{\partial f}{\partial x}(u_x, l_y), \\ u_y, & \text{otherwise.} \end{cases}$$

Lifting from a facet over which the function is convex into a direction in which the function is concave

Let $f : [l, u] \to \mathbf{R}$ be a bivariate function that is convex in x and concave in y and consider the point (x_0, y_0), where $l_x \leq x_0 \leq u_x$ and $y_0 \in \{l_y, u_y\}$. As f is convex when y is fixed, the best linear underestimator is given by the tangent $t : \mathbf{R} \to \mathbf{R}$ on the graph of $f(x, y_0)$ at x_0

$$g(x) \quad := \quad \frac{\partial f}{\partial x}(x_0, y_0)(x - x_0) + f(x_0, y_0).$$

We extend $g(x)$ to a globally valid underestimator of the form $f(x, y) \geq g(x) + \mu(y - y_0)$.

Case (a): $y_0 = l_y$. For every fixed $x \in [l_x, u_x]$ the segment connecting the points $(x, l_y, g(x))$ and $(x, u_y, f(x, u_y))$ underestimates $f(x, y)$ over $[l_y, u_y]$ as $g(x) \leq f(x, l_y)$ and f is concave for every fixed x. The slope of each segment is given by $\gamma(x) = \frac{f(x,u_y)-g(x)}{u_y - l_y}$. A valid lifting coefficient $\mu \in \mathbf{R}$ is the minimal slope $\gamma(x)$ over $x \in [l_x, u_x]$. Note that the function $\gamma(x)$ is convex because $f(x, u_y)$ is convex and $g(x)$ is linear. This means that each critical point $\bar{x}$ satisfying the following first-order condition forms a global minimum of γ:

$$\frac{\partial \gamma}{\partial x}(x) \;=\; \tfrac{1}{u_y - l_y}\Big(\frac{\partial f}{\partial x}(x, u_y) - g'(x)\Big) \quad \stackrel{!}{=} \quad 0.$$

Therefore, $\mu = \frac{f(\bar{x},u_y)-g(\bar{x})}{u_y-l_y}$ as long as a critical point $\bar{x}$ exists which is contained in the domain $[l_x, u_x]$. In case such a point does not exist, set $\bar{x} = l_x$ if $\gamma(l_x) \leq \gamma(u_x)$, and $\bar{x} = u_x$ otherwise.

Case (b): $y_0 = u_y$. Similar to case (a), the segment connecting the points $(x, l_y, f(x, l_y))$ and $(x, u_y, g(x))$ underestimates $f(x, y)$ over $[l_y, u_y]$ for every fixed $x \in [l_x, u_x]$. A valid lifting coefficient $\tau \in \mathbf{R}$ is given by the maximal slope $\gamma(x) = \frac{f(x,l_y)-g(x)}{l_y-u_y}$ over $x \in [l_x, u_x]$. As $l_y - u_y < 0$ and f is convex in x, it follows that γ is concave. Let $\bar{x}$ be a critical point satisfying the first-order condition of $\gamma(x)$, provided such point exists and is contained in $[l_x, u_x]$. If such point does not exist, we set $\bar{x} = l_x$ if $\gamma(l_x) \geq \gamma(u_x)$, and $\bar{x} = u_x$ otherwise. Then, $\tau = \frac{f(\bar{x},l_y)-g(\bar{x})}{l_y-u_y}$.

Lifting from a facet over which the function is convex into a direction in which the function is convex

Let $f : [l, u] \to \mathbf{R}$ be a bivariate function that is strictly convex in both x and y but its Hessian is indefinite. Consider a point $(x_0, y_0) \in [l, u]$ and assume, w.l.o.g., that $x_0 \in \{l_x, u_x\}$ and $l_y \leq y \leq u_y$. As f is convex in y, the best convex underestimator for $f(x_0, y)$ is the function $f(x_0, y)$ itself, i.e., $g(y) = f(l_x, y)$ and $h(y) = f(u_x, y)$. We define the term

$$\mu(x, y) \quad := \quad \frac{f(x, y) - f(x_0, y)}{x - x_0},$$

which is minimized and maximized in Problems (3.11) and (3.12) to compute the best lifting coefficient.

Case (a): $x_0 = l_x$. The best possible lifting coefficient μ corresponds to the infimum of $\mu(x, y)$ over $[l, u]$. As already mentioned in [GHJ+08b], $\mu(x, y)$ is the differential quotient of f in x for fixed y. By convexity of f in x, it follows that $\mu(x, y) \geq \frac{\partial f}{\partial x}(l_x, y)$ for all $(x, y) \in [l, u]$. We can exploit the assumptions on f to show monotonicity of $\frac{\partial f}{\partial x}(l_x, y)$ in y. Formally, the assumptions on f are

- $\frac{\partial^2 f}{\partial x^2}(x, y) > 0$, $\frac{\partial^2 f}{\partial y^2}(x, y) > 0$ for all (x, y) in the interior of $[l, u]$,
- $\frac{\partial^2 f}{\partial x^2}(x, y)\frac{\partial^2 f}{\partial y^2}(x, y) - [\frac{\partial^2 f}{\partial x \partial y}(x, y)]^2 < 0$ for all (x, y) in the interior of $[l, u]$.

Therefore, $[\frac{\partial^2 f}{\partial x \partial y}(x,y)]^2 > \frac{\partial^2 f}{\partial x^2}(x,y)\frac{\partial^2 f}{\partial y^2}(x,y) > 0$ for all (x,y) in the interior of $[l,u]$. As we assume f to be twice continuously differentiable, it follows that $\frac{\partial^2 f}{\partial x \partial y}(x,y)$ is either nonpositive or nonnegative over $[l,u]$ which implies monotonicity of $\frac{\partial f}{\partial x}(l_x,y)$ in y. Thus, $\mu(x,y) \geq \frac{\partial f}{\partial x}(l_x,y) \geq \frac{\partial f}{\partial x}(l_x,\bar{y}) = \mu$ for all $(x,y) \in [l,u]$, where

$$\bar{y} \quad := \quad \begin{cases} l_y, & \text{if } \frac{\partial f}{\partial x}(l_x,l_y) \leq \frac{\partial f}{\partial x}(l_x,u_y), \\ u_y, & \text{otherwise.} \end{cases}$$

Case (b): $x_0 = u_x$. The best possible lifting coefficient τ corresponds to the supremum of $\mu(x,y)$ over $[l,u]$ which is given by $\frac{\partial f}{\partial x}(u_x,\bar{y})$ with

$$\bar{y} \quad := \quad \begin{cases} l_y, & \text{if } \frac{\partial f}{\partial x}(u_x,l_y) \geq \frac{\partial f}{\partial x}(u_x,u_y), \\ u_y, & \text{otherwise.} \end{cases}$$

3.2.3. Computations

We now proceed with a detailed computational study of the described cut-generation algorithm.

Implementation

The cut-generation algorithm is implemented as a new constraint handler in the constraint integer programming framework SCIP [Ach07, Ach09] which has recently been extended to handle general MINLPs [BHV09, Vig12]. SCIP solves MINLPs by a branch-and-bound algorithm. The problem is recursively split into smaller subproblems. In this process a search tree is created and all potential solutions are implicitly enumerated. At each subproblem, standard tools like bound tightening or primal heuristics are employed.

A constraint handler in SCIP defines the semantics and the algorithms to process constraints of a certain class. An enforcement method has to be implemented in each constraint handler, where it is decided whether the optimal solution of the linear relaxation satisfies all of its constraints. If the solution violates one or more constraints, the handler may resolve the infeasibility by adding linear inequalities, performing a domain reduction, or a branching.

Our constraint handler deals with bivariate constraints of the form $\ell \leq f(x, y) + cz \leq r$, where $f : [l, u] \subseteq \mathbf{R}^2 \to \mathbf{R}$ is a bivariate function with fixed convexity behavior, $c \in \mathbf{R}$, $\ell \in \mathbf{R} \cup \{-\infty\}$, and $u \in \mathbf{R} \cup \{\infty\}$. The function $f(x, y)$ has to be passed to the constraint handler in the form of an expression tree (see Section 2.1). Additionally, the convexity behavior of the function has to be specified. Currently, the convexity behavior of bivariate quadratic and monomial functions is recognized automatically according to Table 3.1. However, users can specify the convexity behavior of arbitrary bivariate functions manually using the callable library of SCIP.

For enforcement and during separation rounds the constraint handler generates a linear inequality from under- or overestimators of $f(x, y)$ (as described in the previous sections). If the generated inequality does not cut off the optimal solution of the linear relaxation, spatial branching is applied on either x or y. For instance, if $f(x, y)$ is convex in x and concave in y and the current relaxation's optimum (x_0, y_0, z_0) violates the inequality $f(x_0, y_0) + cz_0 \leq r$, then variable y is proposed as branching candidate to SCIP. From all branching candidates that are registered by all constraint handlers SCIP selects a branching variable and a branching point according to a pseudo-costs based variable selection rule, see [BLL+09, BHV09, Vig12] for details. Further, a feasibility-based bound tightening (FBBT) rule is applied to deduce tighter variable bounds for x, y, or z from the constraint and the bounds on these variables, see [Vig12] for details.

During presolve SCIP reformulates a MINLP into a form which allows to construct a linear relaxation. The reformulation mainly consists of introducing new auxiliary variables and nonlinear constraints for subexpressions of nonlinear functions. For example, a general monomial function $x^p y^q$ has so far been reformulated by SCIP into a product $w_1 w_2$ and two new constraints $w_1 = x^p$ and $w_2 = y^q$ because SCIP knows how to compute linear under- and overestimators for these functions. With the new constraint handler for bivariate functions there is no more need for reformulating monomials $x^p y^q$ with $l_x \geq 0$ and $l_y \geq 0$.

Test set

Initially, we considered the problem libraries GLOBALLib [GLO] and MINLPLib [BDM03]. However, they contain only a few instances with bivariate quadratic terms or monomials, mainly of the form x/y. To investigate the computational benefit of having a convex underestimator

for bivariate functions at hand, we created a set of nonlinear optimization problems, where bivariate functions occur in form of quadratic functions and monomials, e.g., $3x_1^2 + x_1x_2 - x_x^2 + 2x_1^{0.3}x_2^{1.5} - 4x_2^{1.2}x_3^{2.5}$.

The random generation of problems with constraints can lead to infeasibility. As the proposed constraint handler aims at strong bounds on the problem, feasible problems are required in order to compare the quality of the bounds. We designed the following problem class to meet these demands, where we vary the number of variables `Nvars` and constraints `Ncons`, and the maximum degree `Deg` over all constraints:

$$\begin{aligned} &\min \rho \\ &\text{s.t.} \sum_{i=1}^{\texttt{Deg}-1} \sum_{j=1}^{\texttt{Deg}-i} \sum_{k=1}^{\texttt{Nvars}} \sum_{l=k+1}^{\texttt{Nvars}} a_{i,j,k,l,c}\, x_k^{p_{i,j,k,c}}\, x_l^{q_{i,j,l,c}} + \sum_{k=1}^{\texttt{Nvars}} b_{k,c} x_k^2 \le \rho, \\ &\quad \forall\, c \in \{1, \ldots, \texttt{Ncons}\} \quad \text{and} \quad x \in [l, u], \end{aligned}$$

where:

- $p_{i,j,k,c}$: If $i = j = 1$, then $p_{i,j,k,c} = 1$. Otherwise, $p_{i,j,k,c}$ is uniformly random set to a value in $\{(i-1)+0.2, (i-1)+0.4, \ldots, (i-1)+1\}$.
- $q_{i,j,l,c}$: If $i = j = 1$, then $q_{i,j,l,c} = 1$. Otherwise, $q_{i,j,l,c}$ is uniformly random set to a value in $\{(j-1)+0.2, (j-1)+0.4, \ldots, (j-1)+1\}$.
- $a_{i,j,k,l,c}$: If $i = j = 1$, k odd, and $l = k+1$, then $a_{i,j,k,l,c}$ is uniformly random in $\{-4, -3, \ldots, 3, 4\}$. If $i > 1$ or $j > 1$, then $a_{i,j,k,l,c}$ is with probability $2/\texttt{Nvars}$ in $\{-4, -3, \ldots, 3, 4\}$. Otherwise, we set $a_{i,j,k,l,c} = 0$.
- $b_{k,c}$: The coefficient $b_{k,c}$ is chosen uniformly at random from the set $\{-4, -3, \ldots, 3, 4\}$.
- $[l, u]$: For each $k \in \{1, \ldots, \texttt{Nvars}\}$ the lower bound l_k is uniformly at random set to a value in $\{0, 1, 2, 3, 4\}$. The upper bound u_k is the sum of $l_k + 1$ and a value which is chosen uniformly at random from $\{0, 1, 2, 3, 4\}$. To avoid numerical inconsistencies, we check that $u_k^{\texttt{Deg}} \le 2000$.

The condition $i = j = 1$ deals with the quadratic case. It ensures that integer exponents leading to quadratic terms x_ix_j are generated. The condition $(l = k+1)$, l odd, leads to bivariate quadratic terms $x_1x_2, x_3x_4, \ldots$

and thus, the univariate quadratic terms x_k^2 can be associated to a unique bivariate quadratic monomial.

The implemented methods are of particular interest if the optimal or intermediate solutions are attained in the interior of the underlying boxes $[l, u]$. Otherwise, only the lifting procedures are executed. Thus, the following ellipsoid constraint is optionally added to the problems which cuts off the boundary of the box

$$\sum_{k=1}^{\texttt{Nvars}} \left(\frac{x_k - \text{midpoint}_k}{\text{interval-length}_k/2} \right)^2 = \sum_{k=1}^{\texttt{Nvars}} \left(\frac{x_k - (u_k + l_k)/2}{(u_k - l_k)/2} \right)^2 \leq 1. \tag{3.13}$$

The following settings are considered:

- `Nvars` $\in \{10, 20, 30\}$,
- `Deg` $\in \{2, 3, 4, 5\}$,
- `Ncons` $\in \{1, 2, 3, 5, 10\}$,
- Enable/Disable the ellipsoid constraint (3.13).

Hence, there are $3 \cdot 5 \cdot 4 \cdot 2 = 120$ different settings. For each setting we generate 10 random instances leading to 1,200 instances in total.

Experimental Setup

We compared SCIP 3.0.0 (with the new constraint handler enabled or disabled) with BARON 11.1.0 [TS05] and COUENNE 0.4 [BLL+09]. SCIP and BARON use CPLEX 12.4 for solving LP relaxations, COUENNE uses CLP 1.14. SCIP and COUENNE use Ipopt 3.10 for finding local optimal solutions to an NLP, BARON uses MINOS 5.51.

We run all experiments under openSuSE Linux 11.4 64bit on a Dell PowerEdge M1000e blade with 48 GB RAM and two Intel Xeon X5672 CPUs running at 3.20 GHz. The timelimit is 30 minutes and the gap tolerance is 0.01%.

Results

In Table 3.2 the results for performing all 1,200 instances by BARON, COUENNE, SCIP, and SCIP(bivar) with the new constraint handler enabled are summarized, where we exclude 17 instances for which one of the 4 algorithms aborted or failed. First, we report the number of instances which are solved and solved fastest by an algorithm, and for which an algorithm computes the best dual bound. An algorithm is marked fastest

	SCIP	SCIP(bivar)	BARON	COUENNE
#solved	369	**966**	643	711
#fastest	34	**625**	183	208
#best dual bound	370	**1026**	681	814
time (sh. geom. mean)	499.6	**75.0**	266.2	191.6
nodes (sh. geom. mean)	2166.7	**391.6**	1373.6	3818.2
dual gap (arith. mean)	30.07%	**5.62%**	20.31%	18.35%

Table 3.2.: Computational results for 1,200 randomly generated polynomial instances.

if it is within one second of the minimal solution time for an instance. Similarly, a dual bound for a solver is marked as best dual bound, if the bound is within 0.01% of the best dual bound for that instance. The definitions of fastest algorithm and best dual bound imply that two algorithms can be marked fastest/best for one instance. Second, for each solver we calculated mean values of the solution time in which unsolved instances are accounted for with the time limit, the number of processed nodes, and the dual gap at termination. The mean values are computed according to [Ach07, Section A.3], where the shifted geometric mean, defined as

$$\left(\prod_{i\in[n]} \max(\varepsilon, v_i + s)\right)^{1/n} - s,$$

is calculated with $\varepsilon = 1$ and $s = 10$ for solution times and with $\varepsilon = 1$ and $s = 100$ for node counts. The *dual gap* [ABH12] for a problem with dual bound $\underline{v}$ and best known objective value v^* is defined as

$$\text{dual gap} := \min\left(1, \frac{|v^* - \underline{v}|}{\max(1, |v^*|)}\right).$$

Bounding the gap from above by 1 reduces the impact of outliers and results in meaningful arithmetic means.

The results in Table 3.2 allow for an overall ranking of the four algorithms as the tendency is the same for all the individual performance parameters. SCIP(bivar) clearly outperforms the other algorithms, followed by COUENNE, BARON, and SCIP. SCIP(bivar) solves at least 200

instances more than the solvers BARON and COUENNE, which is also a reason why its dual gap and its solution time is much better than the ones of BARON and COUENNE. As SCIP performs worst, the results indicate that the use of the new constraint handler within BARON or COUENNE may even lead to better results.

To exclude the influence of unsolved instances, we restrict our attention to the 529 instances which are solved by SCIP(bivar), BARON as well as COUENNE. Table 3.3 depicts the results and shows that SCIP(bivar) is the fastest algorithm for most instances. The mean of the computation times shows that COUENNE is almost as fast as SCIP(bivar), but that BARON needs two times more CPU time than SCIP(bivar). A possible explanation for the speed of SCIP(bivar) is the low number of nodes processed indicating the strength of the relaxations used in SCIP(bivar). This claim is supported by the direct comparison of SCIP and SCIP(bivar) restricted to the 357 instances solved by both algorithms. SCIP exhibits a mean of 19.7 seconds and 1276.4 nodes while SCIP(bivar) uses only 11.3 seconds and 449.4 nodes. Thus, SCIP(bivar) can utilize the improved relaxations to avoid branching steps and to prune nodes earlier, thus accelerating the computations.

	SCIP(bivar)	BARON	COUENNE
#fastest	**235**	170	125
time (sh. geom. mean)	**16.5**	30.5	18.3
nodes (sh. geom. mean)	**390.3**	485.7	550.7

Table 3.3.: Summary of 529 instances solved by SCIP(bivar), BARON as well as COUENNE.

In Figure 3.8 we refine the analysis of the dual gaps for the 1,200 instances w.r.t. the number of variables `NVars`, the number of constraints `Ncons`, the maximal degree `Deg` of the polynomials, and the ellipsoid constraint in Equation (3.13) dis- and enabled. Regarding the number of variables, we observe that for 10 variables per instance SCIP(bivar), BARON, and COUENNE have about the same dual gap close to zero. For instances with more variables the dual gaps of BARON and COUENNE grow tremendously up to a gap of more than 34% while the dual gap of SCIP(bivar) increases modestly to 13%.

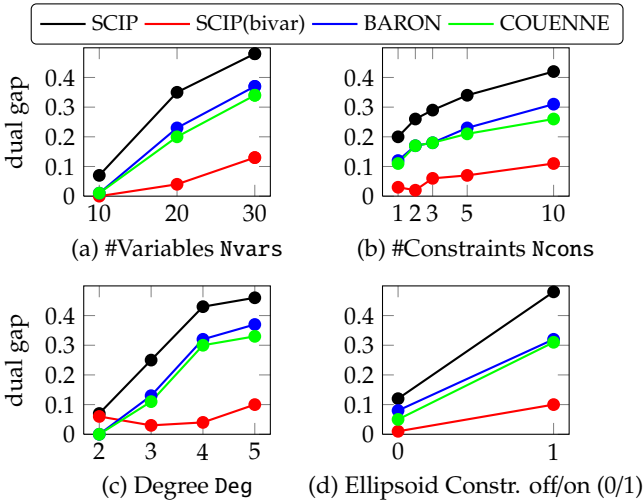


Figure 3.8.: Dual gaps (arithm. mean) of the solvers w.r.t. the number of variables `NVars`, constraints `Ncons`, the maximal degree `Deg`, and the ellipsoid constraint dis- and enabled.

An increase in the number of constraints leads only to a modest increase in the dual gap for all solvers. SCIP(bivar)'s dual gap is about 2.5-7 times better than the second best algorithm COUENNE.

In contrast to the number of constraints, the maximal degree has a significant influence on the dual gap of all solvers. For a degree of two BARON's and COUENNE's dual gaps are close to zero while SCIP(bivar)'s gap is about 6%. For degrees of three and four the dual gaps of BARON and SCIP increase heavily whereas SCIP(bivar)'s gap even decreases. The reason for this can be found in the construction of the instances. For degree two we construct only bivariate quadratic monomials like xy but no monomials with fractional exponents like $x^{0.2}y^{1.4}$ which is allowed for larger degrees. As a consequence the programs corresponding to degree two are quadratic programs for which solvers like BARON and COUENNE apply well-suited, pre-defined relaxations while SCIP(bivar) solves a series of auxiliary problems to compute linear under- and overestimators.

For instances with larger degrees there are monomials like $x^{0.4}y^{1.4}$, which are concave in one variable and convex in the other variable, and monomials like $x^{1.4}y^{2.2}$, which are 1-convex and indefinite. These cases are handled by the presented constraint handler in SCIP(bivar). A proper relaxation of these terms might compensate the more time consuming relaxation of the bivariate quadratic terms which helps to explain the smaller dual gap of SCIP(bivar) for degree 3 and 4.

The enabled ellipsoid constraint leads to an increase in the dual gaps of SCIP, SCIP(bivar), BARON, and COUENNE by factors of about 4, 10, 4, and 6, respectively. The activation of the ellipsoid constraint forces the optimal solution to be attained in the interior of the given domains which obviously causes some problems for the algorithms due to weaker relaxations. Yet, SCIP(bivar) returns a dual gap which is at least three times better than the gap of the other algorithms. This shows that both the lifting technique used to cut-off points at the boundary and the linear underestimators based on the convex envelopes used to cut-off points in the interior have a significant influence on the performance of SCIP(bivar).

In all tests so far we constructed monomials with fractional exponents like $x^{0.4}y^{1.8}$. A last comparison is now devoted to instances with integral exponents only, i.e., we round up the exponents such that we obtain monomials like x^1y^2. We compare the computational results of SCIP and SCIP(bivar) applied to a test set of 1,200 instances with integral exponents in Table 3.4. Compared to the instances with fractional exponents, presented in Table 3.2, SCIP can solve about 300 instances more whereas SCIP(bivar) solves about 300 instances less.

	SCIP	SCIP(bivar)
#solved	**667**	612
#fastest	**497**	194
#best dual bound	**1075**	734
time (sh. geom. mean)	**205.2**	254.6
nodes (sh. geom. mean)	1802.9	**1039.9**
dual gap (arith. mean)	**22.57%**	26.30%

Table 3.4.: Computational results for 1,200 randomly generated polynomial instances with integral exponents.

To understand the clear improvement of SCIP, consider the monomials

$x^{0.4}y^{1.8}$ and $x^{2.6}y^{1.6}$. SCIP introduces new variables $v_{0.4}, v_{2.6}$, and $w_{2.6}, w_{1.6}$ for the univariate convex or concave monomials $x^{0.4}, x^{2.6}$, and $y^{1.8}, y^{1.6}$, respectively. Afterwards, it relaxes the univariate monomials and the bilinear product terms $v_{0.4}w_{1.8}$ and $v_{1.8}w_{1.6}$ by their convex and concave envelopes. If only integral exponents are allowed, these monomials now read x^1y^2 and x^3y^2. Thus, less variables are introduced, namely v_1, v_3, and w_2, and the bilinear terms v_1w_2 and v_3w_2 have a common variable which is helpful in the process of relaxation.

A possible explanation for SCIP(bivar)'s bad performance is the nonoccurrence of monomials like $x^{0.4}y^{1.8}$ which are strictly concave in one variable and convex in the other one. We already indicated this in the discussion of Figure 3.8 (c), where no monomials like $x^{0.4}y^{1.8}$ occur for degree two while this is the case for higher degrees. A further indication for this conjecture is given in Table 3.5, where the dual gaps for SCIP and SCIP(bivar) w.r.t. the maximal degree of the programs with fractional and integral exponents are displayed. The numbers show an enormous increase in the dual gap of SCIP(bivar) for integral exponents compared to fractional exponents. Note that the instances with integral exponents also contain indefinite (n-1)-convex monomial functions like x^3y^2 which are also covered by the new constraint handler. From the bad performance of SCIP(bivar) we infer that the computation of the related cuts is not yet efficient.

	Max degree	2	3	4	5
SCIP	Fractional	7.00%	26.83%	49.79%	52.00%
	Integral	7.00%	18.88%	31.55%	32.84%
SCIP(bivar)	Fractional	6.25%	3.33%	9.28%	14.54%
	Integral	6.26%	23.72%	35.84%	39.36%

Table 3.5.: Dual gaps of SCIP and SCIP(bivar) for instances with fractional exponents and instances with integral exponents.

To sum up, the new constraint handler used in SCIP(bivar) can reduce the solution time and improve the dual bounds of programs containing bivariate functions with a fixed convexity behavior. Excellent results are obtained if the functions are strictly concave in one direction and convex in the other direction. In these cases the algorithm SCIP(bivar) clearly

outperforms standard algorithms w.r.t. dual bounds and running time.

An advantage of the presented constraint handler is its applicability to general bivariate functions with a fixed convexity behavior. However, the additional incorporation of explicit formulas for known convex envelopes can help to reduce the computation of auxiliary problems in the cut-generation algorithm. Examples are fractional terms from Example 3.20, bivariate quadratic terms from Example 3.21, and the recently derived envelopes for functions $f(x, y) = g(x)h(y)$, where $g(x)$ is an univariate concave function and $h(y)$ is an univariate convex function of the form y^a or e^y (see Section 3.1.3). To handle more general bivariate functions by the new constraint handler, the automatic detection of the convexity behavior needs to be extended to further classes of functions in the future.

3.3. Chromatographic Processes with Second-Order Isotherms

In the last section of this chapter we investigate novel concepts in chromatographic processes which are frequently used separation processes in the biotechnology, the pharmaceutical, and the petrochemical industry. The separation is achieved by passing a dissolved multicomponent mixture in a mobile (liquid) phase through a stationary (solid) phase. As a result of the different adsorption properties of single components towards the stationary phase, the desired components are isolated.

A crucial part in the analysis and design of chromatographic processes is to describe the adsorption behavior between the two phases by so-called *isotherms*, i.e., equilibrium functions that reflect the relation between the concentration of the components in the solid- and in the liquid-phase. For rather simplified isotherms there are nowadays reliable design rules to decide whether a separation is feasible and which operating parameters should be used. These isotherms are, however, not sufficient for many applications, e.g., they do not allow to model inflection points in the course of adsorption, a phenomena which is frequently observed for more realistic adsorption isotherms. For more complex equilibrium functions theoretical methods for the design of chromatographic processes are less developed. In this work we analyze the design of chromatographic processes with *second-order isotherms* which are capable to describe inflection

points and are given by

$$f(x_1, x_2) = \frac{q_s x_1 (b_{1,0} + 2b_{2,0}x_1 + b_{1,1}x_2)}{1 + b_{1,0}x_1 + b_{0,1}x_2 + b_{2,0}x_1^2 + b_{1,1}x_1x_2 + b_{0,2}x_2^2},$$

with nonnegative coefficients $q_S, b_{1,0}, b_{0,1}, b_{2,0}, b_{1,1}, b_{0,2}$ and variables x_i reflecting the liquid-phase concentrations of component i.

In Section 3.3.1 we present the process engineering background and a mathematical model for chromatographic processes. We briefly introduce the conventional isotherm models and discuss the design rules for the corresponding chromatographic processes.

Motivated by the rare information for processes with more complicated isotherms, we investigate the behavior of chromatographic processes based on second-order isotherms in this section. On the one hand, a classical scanning technique is used to identify the region of applicable operating parameters. On the other hand, an alternative approach is suggested which verifies the existence and shape of the suitable parameter region by infeasibility certificates. For this, we study several relaxation strategies for second-order isotherms and computationally compare their strength in Section 3.3.2. The relaxations are then used to determine the shape of the separation regions of chromatographic processes with second-order isotherms in Section 3.3.3.

This section is based on [BMSMW10] which extends the results in [HMSMW07], where chromatographic processes with linear isotherms are analyzed w.r.t. feasibility. Keep in mind that the paper was published in 2010, so that the software and hardware used in this section have to be put into the corresponding context.

3.3.1. Fundamentals of Chromatographic Processes

Continuous counter-current chromatographic processes can be well described by the simplifying true moving bed (TMB) model [RPM09, RC89]. In this section we initially present the model formulation for TMB processes used in this work. Then, adsorption models for isotherms are presented and the corresponding separation regions are discussed.

The Principle of TMB Processes

In a chromatographic process the separation of a binary mixture $A + B$ is based on a different adsorption behavior of the components A and B

w.r.t. a certain solid. We assume that component A is the less adsorbable component while B denotes the more adsorbable component. A continuously operated counter-current chromatographic unit consists of four zones separated by two inlet plates (F), (D) and two outlet plates (R), (E) (cf. Figure 3.9). Each zone i is further subdivided into a theoretical number of plates M_i.

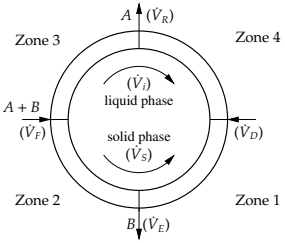


Figure 3.9.: A sketch of a true moving bed (TMB) process.

The separation of the mixture is achieved by a continuous counter-current movement of a *liquid-phase* and a *solid-phase* through the system. The binary mixture $A + B$ is fed into the TMB unit at the feed plate (F). A second inlet plate (D) is used to feed a solvent into the system. At the outlet plates (R) and (E) the single components A and B (or enriched streams), respectively, are withdrawn. The separation of the mixture takes place in zones 2 and 3. The more adsorbable component B is enriched in zone 2 while the less adsorbable component A is enriched in zone 3. In zones 1 and 4 the solid and the solvent are purified, respectively. For a continuous counter-current chromatographic process it is assumed that the solid-phase moves continuously through the entire system with a flow-rate $\dot{V}_s$ while the *internal volumetric liquid-phase* flow-rates $\dot{V}_i$, $i \in \{1, 2, 3, 4\}$, may differ between distinct zones. There are four *external volumetric liquid-phase* flow-rates $\dot{V}_j$, $j \in \{\text{F,R,D,E}\}$, which are linked to the chromatographic system by the inlet and outlet plates. It is convenient to model the process in terms of four dimensionless ratios of the liquid-phase flow-rates $\dot{V}_i$ with respect to the solid-phase flow-rate $\dot{V}_s$ [MMM98, SMMC93]:

$$m_i \quad := \quad \frac{\dot{V}_i}{\dot{V}_s}, \qquad \text{for all } i \in \{1, 2, 3, 4\}.$$

The relations between the internal and the external flow-rate ratios are given by the following linear system of equations:

$$m_F = m_3 - m_2, \quad m_R = m_3 - m_4, \quad m_D = m_1 - m_4, \quad m_E = m_1 - m_2. \tag{3.14}$$

On every plate each component k is present with a certain concentration in both the liquid-phase and the solid-phase. In the following we denote by $c^k_{i,j}$ ($q^k_{i,j}$) the liquid-phase (solid-phase) concentrations of the kth component, $k \in \{A, B\}$, in the jth plate of zone i, with $j \in \{0, \ldots, M_i\}$. The relation between the liquid-phase and solid-phase concentrations $c^k_{i,j}$, $q^k_{i,j}$ is described by isotherms, i.e., $q^k_{i,j}$ can be seen as a function $q^k_{i,j} : \mathbf{R}^2 \to \mathbf{R}$, $(c^A_{i,j}, c^B_{i,j}) \mapsto q^k_{i,j}(c^A_{i,j}, c^B_{i,j})$. Special classes of isotherms are discussed in more detail later in this section. The variables $c^k_{i,0}$ ($q^k_{i,0}$) represent the liquid-phase (solid-phase) concentrations of the kth component in the plate connecting zone i with the previous zone via plates (E), (F), (D), or (R), respectively. The concentration of component k in the external liquid-phase streams are abbreviated by c^k_j, $j \in \{\mathrm{E}, \mathrm{F}, \mathrm{D}, \mathrm{R}\}$, where c^k_F and c^k_D are input parameters and $c^k_{2,0} = c^k_E$ and $c^k_{4,0} = c^k_R$ are the output of the model.

To model a TMB unit, the classical equilibrium stage model is used dividing the unit in a discrete number of theoretical plates (cf. [BHSM03, RC89]). For all plates the steady state *mass balance equations* must be fulfilled:

$$0 = \begin{cases} q^k_{i,j+1} + m_i c^k_{i,j-1} - m_i c^k_{i,j} - q^k_{i,j}, & \text{if } j = 1, \ldots, M_i - 1, \\ q^k_{i+1,0} + m_i c^k_{i,M_i-1} - m_i c^k_{i,M_i} - q^k_{i,M_i}, & \text{if } j = M_i. \end{cases} \tag{3.15}$$

Using the expression for the external flow-rate rations in Equation (3.14), the mass balance equations for the inlet and outlet plates result in

$$\begin{array}{lll} (F) & q^k_{3,1} + m_2 c^k_{2,M_2} - m_3 c^k_{3,0} - q^k_{3,0} + (m_3 - m_2) c^k_F & = 0, \\ (R) & q^k_{4,1} + m_3 c^k_{3,M_3} - m_4 c^k_{4,0} - q^k_{4,0} - (m_3 - m_4) c^k_{4,0} & = 0, \\ (D) & q^k_{1,1} + m_4 c^k_{4,M_4} - m_1 c^k_{1,0} - q^k_{1,0} + (m_1 - m_4) c^k_D & = 0, \\ (E) & q^k_{2,1} + m_1 c^k_{1,M_1} - m_2 c^k_{2,0} - q^k_{2,0} - (m_1 - m_2) c^k_{2,0} & = 0. \end{array} \tag{3.16}$$

Moreover, the total mass balance equation concerning the overall mass conservation must be satisfied:

$$0 = (m_3 - m_4) c^k_{4,0} + (m_1 - m_2) c^k_{2,0} - (m_1 - m_4) c^k_D - (m_3 - m_2) c^k_F. \tag{3.17}$$

The goal of the separation process is to produce the components with a certain purity $\text{pur}^k \in [0,1]$ which is reflected by the purity requirements:

$$\frac{c_R^A}{c_R^A + c_R^B} \geq \text{pur}^A, \qquad \frac{c_E^B}{c_E^A + c_E^B} \geq \text{pur}^B. \tag{3.18}$$

Isotherms Based on Statistical Thermodynamics

A main aspect of modeling a chromatographic separation process is to find a suitable description of the relation between the liquid-phase concentrations $c_{i,j}^k$ and the corresponding solid-phase concentrations $q_{i,j}^k$ for equilibrium conditions. A simple way to model such a relation is to use *linear isotherms* which assume a linear relationship, i.e.,

$$q_{i,j}^k = H^k \, c_{i,j}^k, \quad k \in \{A, B\},$$

where H^k stands for the *Henry constant* of the kth component that reflects its adsorption behavior with respect to the solid. Linear isotherms are often used to describe the equilibrium behavior of different sugars (see, e.g., [CCHU92]). In contrast to linear isotherms, *Langmuir isotherms* can reflect the often observed competitive behavior between the components [GFSK06]:

$$q_{i,j}^A = \frac{q_s c_{i,j}^A \, b_{1,0}}{1 + b_{1,0} c_{i,j}^A + b_{0,1} c_{i,j}^B}, \qquad q_{i,j}^B = \frac{q_s c_{i,j}^B \, b_{0,1}}{1 + b_{1,0} c_{i,j}^A + b_{0,1} c_{i,j}^B}.$$

The Langmuir isotherm model can be considered as a special case of more general isotherm models that can be derived from *statistical thermodynamics*. One possibility to model isotherms based on statistical thermodynamics is given by

$$q_{i,j}^k : \mathbf{R}^2 \to \mathbf{R}, \quad (c_{i,j}^A, c_{i,j}^B) \mapsto \frac{q_s c_{i,j}^k \, \frac{\partial P}{\partial c_{i,j}^k}(c_{i,j}^A, c_{i,j}^B)}{P(c_{i,j}^A, c_{i,j}^B)}, \qquad k \in \{A, B\},$$

where P is a real polynomial of degree d with constant term equal to one [Hil60]. The higher the degree of the polynomial, the more phenomena of the adsorption behavior can be modeled. Langmuir isotherms are statistical isotherms of degree one.

In this section we focus on statistical isotherms, where P is of degree

two ($d = 2$), so-called *second-order isotherms*. For a binary mixture the isotherms appear as

$$\begin{aligned} q_{i,j}^A &= \frac{q_s c_{i,j}^A \left(b_{1,0} + 2b_{2,0}c_{i,j}^A + b_{1,1}c_{i,j}^B\right)}{1 + b_{1,0}c_{i,j}^A + b_{0,1}c_{i,j}^B + b_{2,0}\left(c_{i,j}^A\right)^2 + b_{1,1}c_{i,j}^A c_{i,j}^B + b_{0,2}\left(c_{i,j}^B\right)^2}, \\ q_{i,j}^B &= \frac{q_s c_{i,j}^B \left(b_{0,1} + 2b_{0,2}c_{i,j}^B + b_{1,1}c_{i,j}^A\right)}{1 + b_{1,0}c_{i,j}^A + b_{0,1}c_{i,j}^B + b_{2,0}\left(c_{i,j}^A\right)^2 + b_{1,1}c_{i,j}^A c_{i,j}^B + b_{0,2}\left(c_{i,j}^B\right)^2}. \end{aligned} \tag{3.19}$$

Compared to linear and Langmuir isotherms, second-order isotherms are capable of describing inflection points that are frequently encountered in real systems [GMNS60, DG91, ZSSM06, RPM09].

Separation regions

An important question in the design of continuous counter-current chromatographic processes is to find suitable values of the dimensionless flow-rate ratios m_i. Significant contributions in this direction have been made for linear and Langmuir isotherms ([Maz06, SMMC93]). For linear isotherms, complete separation ($\text{pur}^k = 100\%$), and an infinite number of plates, Storti et al. ([SMMC93]) developed the so-called *triangle theory*: For given values of m_1 and m_4 the region for m_2 and m_3 allowing successful separation is shaped like a triangle as illustrated in Figure 3.10 (a). Migliorini et al. [MMM98] extended this result to Langmuir isotherms, where a 'triangle'-like separation region can be determined analytically, cf. Figure 3.10 (b). Mazzotti [Maz06] presented an extended equilibrium theory based analysis for a generalized isotherm model capable to describe convex and concave (Langmuir and anti-Langmuir) behavior.

Note that analytical solutions for a complete separation region as summarized above can be derived only by assuming an infinite number of plates. In practice one has to carry out separation processes with a finite number of plates under specific (reduced) purity requirements, for which analytical solutions are not known. A common approach to determine the shape of the separation region is to apply a simple *scanning technique*. The idea is to fix the key variables of the process, namely the four flow-rate ratios m_i, to points of a pre-defined grid. Then, numerical methods are applied to search for a solution in the remaining variables for every

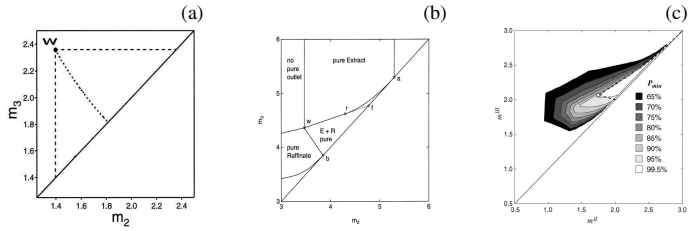


Figure 3.10.: Complete separation regions for (a) linear and (b) Langmuir isotherms [MMM98]. (c) Separation regions for Langmuir isotherms under reduced purity requirements [KSMK07].

discrete grid point. Figure (3.10) (c) displays the results by Kaspereit et al. [KSMK07] using a scanning technique for processes with Langmuir isotherms under reduced purity requirements.

In the past scanning techniques were frequently used to analyze continuous counter-current chromatographic processes with linear and Langmuir isotherms (cf. [BHSM03, KSMK07]. Up to now, the explicit incorporation of single solute and competitive isotherms exhibiting inflection points has been rarely considered in both theoretical and computational studies (cf. [MSMG04, RPM09]) which motivated the work in this section.

3.3.2. Relaxation of Second-Order Isotherms

Besides the scanning technique, we also apply a relaxation technique based on ideas from global optimization (see, e.g., [HMSMW07, McC76, TS04, BL12]) in order to determine the separation region. In contrast to the scanning technique, the alternative relaxation approach does not focus on the feasibility of a certain point but on the infeasibility of subdomains. Therefore, this approach will lead to the negative image of the results obtained by the scanning technique described above.

While feasibility of a point can be checked easily, it is generally hard to prove infeasibility of a problem. However, if a problem is given by a system of linear equations and/or inequalities, certificates are available to prove infeasibility, e.g., the Farkas Lemma (cf. [Roc70]). In order to construct a linear relaxation of the TMB model, we follow a standard approach and substitute each nonlinear term by a new variable which is

under- and overestimated by affine hyperplanes.

The TMB model in Section 3.3.1 consists of two classes of nonlinearities: (i) the bilinear terms $m \cdot c_{i,j}^k$ in the mass balance Equations (3.15), (3.16), and (3.17), and (ii) the second-order isotherms in Equation (3.19). The bilinear terms can be relaxed best possible by their convex and concave envelopes presented in Example 3.14. To the best of our knowledge, the envelopes of second-order isotherms are not known. Our aim is to analyze certain relaxation strategies for second-order isotherms in order to determine strong convex under- and concave overestimators. It is worthwhile to study such under- and overestimators as second-order isotherms exhibit a very general structure as a quotient of two bivariate polynomials. Note that Langmuir isotherms are a special case of second-order isotherms for which a characterization of the envelopes is at hand (cf. [JKMW08, TS01]).

We investigate four distinct relaxation strategies. The first three strategies are based on a reformulation of the isotherms into structures for which the convex envelope is known while the fourth strategy exploits the lifting technique presented in Section 3.2.2. Note that the implementation of the first three strategies was already documented in [Bal08]. To illustrate the relaxation strategies, we consider the isotherm for component A from Equation (3.19) omitting some indices

$$q^A(c^A, c^B) = \frac{q_s\, c^A \left(b_{1,0} + 2b_{2,0}c^A + b_{1,1}c^B\right)}{1 + b_{1,0}c^A + b_{0,1}c^B + b_{2,0}\,(c^A)^2 + b_{1,1}c^Ac^B + b_{0,2}\,(c^B)^2} =: \frac{r(c^A, c^B)}{s(c^A, c^B)}.$$

Relaxation Strategy One (RS1): We multiply the expression by the denominator and expand the terms to

$$\begin{aligned}&\mathbf{q^A} + b_{1,0}\mathbf{q^Ac^A} + b_{0,1}\mathbf{q^Ac^B} + b_{2,0}\mathbf{q^A}(\mathbf{c^A})^2 + b_{1,1}\mathbf{q^Ac^Ac^B} + b_{0,2}\mathbf{q^A}(\mathbf{c^B})^2\\ &= q_s b_{1,0}\mathbf{c^A} + 2q_s b_{2,0}(\mathbf{c^A})^2 + q_s b_{1,1}\mathbf{c^Ac^B},\end{aligned}$$

where the variables are given in bold. The convex terms $(c^A)^2$ and $(c^B)^2$ are substituted by new variables and relaxed by their envelopes. Then, only bilinear terms like q^Ac^A and trilinear terms like $q^Ac^Ac^B$ remain which are relaxed by their envelopes as given in Examples 3.14 and 3.15, respectively.

Relaxation Strategy Two (RS2): We multiply the expression by the de-

nominator but do not expand the terms such that we obtain

$$q^A s' = r', \quad s' = s(c^A, c^B), \quad r' = r(c^A, c^B).$$

The bilinear term $q^A s'$ and the bivariate quadratic terms $s(c^A, c^B)$ and $r(c^A, c^B)$ are relaxed by their envelopes as given in Examples 3.14 and 3.21, respectively.

Relaxation Strategy Three (RS3): We do not multiply the expression by the denominator and relax the isotherms by

$$q^A = \frac{r'}{s'}, \quad s' = s(c^A, c^B), \quad r' = r(c^A, c^B).$$

The fractional term $\frac{r'}{s'}$ and the bivariate quadratic terms $s(c^A, c^B)$ and $r(c^A, c^B)$ are relaxed by their envelopes as given in Examples 3.20 and 3.21, respectively.

Relaxation Strategy Four (RS4): Instead of performing the reformulation step that requires the introduction of additional variables we apply the lifting technique (see Section 3.2.2) directly to the isotherms to relax them. Assume we want to determine an underestimator for $q^A(c^A, c^B)$ over the domain $[l^A, u^A] \times [l^B, u^B]$. The first step is to fix one of the variables to its lower or upper bound, e.g., $c^B = l^B$. Obviously, the term $q^A(c^A, l^B)$ is an underestimator for $q^A(c^A, c^B)$ along the line $c^B = l^B$. To extend this underestimator to the entire box, we need a lifting coefficient $\mu \in \mathbf{R}$ such that $q^A(c^A, c^B) \geq q^A(c^A, l^B) + \mu(c^B - l^B)$ for all $(c^A, c^B) \in [l^A, u^A] \times [l^B, u^B]$, or equivalently,

$$\mu \quad \leq \quad \inf \left\{ \underbrace{\frac{q^A(c^A, c^B) - q^A(c^A, l^B)}{c^B - l^B}}_{=:\mu(c^A, c^B)} \;\middle|\; (c^A, c^B) \in [l^A, u^A] \times (l^B, u^B] \right\} \;=:\; \mu^*. \tag{3.20}$$

The best possible lifting coefficient is given by $\mu = \mu^*$.

Proposition 3.26. *Let $\mu(c^A, c^B)$ be monotonously decreasing in c^A over a given domain $[l^A, u^A] \times [l^B, u^B]$. Then,*

$$\mu^* = \min \left\{ \frac{\partial q^A}{\partial c^B}(u^A, l^B),\ \mu(u^A, (c^B)^+),\ \mu(u^A, (c^B)^-),\ \mu(u^A, u^B) \right\},$$

where $(c^B)^+$ and $(c^B)^-$ can be uniquely determined as roots of a quadratic equation

corresponding to the numerator of $\frac{\partial \mu}{\partial c^B}(u^A, (c^B)^{\pm})$.

Proof. As the function $\mu(c^A, c^B)$ is monotonously decreasing in c^A, it holds that $\mu(c^A, c^B) \geq \mu(u^A, c^B)$ for all (c^A, c^B) in the underlying domain. Thus, the optimal c^B is either attained at (i) the boundary of the interval $[l^B, u^B]$, or (ii) it is a root of $\frac{\partial \mu}{\partial c^B}(u^A, c^B)$. For case (i) we have that $\mu(u^A, c^B) \to \frac{\partial q^A}{\partial c^B}(u^A, l^B)$ if $c^B \to l^B$. For case (ii) one can check that the numerator of $\frac{\partial \mu}{\partial c^B}(u^A, (c^B)^{\pm})$ is given by $q_S u^A (c^B - l^B)^2 (a_2 (c^B)^2 + a_1 c^B + a_0)$, where a_2, a_1, and a_0 are constants depending on $u^A, l^B, b_{1,0}, b_{0,1}, b_{2,0}, b_{0,2}$, and $b_{1,1}$. If $a_2 \neq 0$, which is the general case in our setting as $a_2 = b_{1,0} b_{0,2}^2 + 2 b_{2,0} u^A b_{0,2}^2 + b_{1,1} l^B b_{0,2}^2$, the solution of the quadratic equation is given by $(c^B)^{\pm} = \frac{-a_1 \pm \sqrt{a_1^2 - 4 a_2 a_0}}{2 a_2}$. □

The resulting underestimator $q^A(c^A, l^B) + \mu^*(c^B - l^B)$ is not necessarily convex as $q^A(c^A, l^B)$ is not necessarily convex. With the help of the next statement, we can apply standard analysis to compute its convex envelope (e.g., see [McC76]).

Proposition 3.27. *The univariate function* $q^A(c^A, l^B)$ *restricted to a nonnegative interval* $[l^A, u^A] \subseteq \mathbf{R}_{\geq 0}$ *is either convex, concave, or it is first convex and then concave.*

Proof. To check convexity of $q^A(c^A, l^B)$ w.r.t. c^A, we consider $\frac{\partial^2 q^A(c^A, l^B)}{\partial (c^A)^2}$. The denominator of this term is given by $(1 + b_{1,0} c^A + b_{0,1} l^B + b_{2,0} (c^A)^2 + b_{1,1} c^A l^B + b_{0,2} (l^B)^2)^3$ and it is positive as all involved variables and coefficients are nonnegative. The numerator is a cubic function of the form $-2 q_S (a_3 (c^A)^3 + a_2 (c^A)^2 + a_1 c^A + a_0)$, where

$$
\begin{aligned}
a_3 :=& b_{1,0} b_{2,0}^2 + b_{2,0}^2 b_{1,1} l^B, \\
a_2 :=& 6 b_{0,1} l^B b_{2,0}^2 + 6 b_{2,0}^2 + 6 b_{2,0}^2 b_{0,2} (l^B)^2, \\
a_1 :=& 3 b_{2,0} b_{1,1} (l^B)^3 b_{0,2} + (3 b_{2,0} b_{0,1} b_{1,1} + 3 b_{2,0} b_{1,0} b_{0,2}) (l^B)^2 \\
&+ (3 b_{2,0} b_{1,1} + 3 b_{2,0} b_{1,0} b_{0,1}) l^B + 3 b_{2,0} b_{1,0}, \\
a_0 :=& (-2 b_{2,0} b_{0,2}^2 + b_{1,1}^2 b_{0,2}) (l^B)^4 + (b_{1,1}^2 b_{0,1} + 2 b_{1,0} b_{1,1} b_{0,2} - 4 b_{2,0} b_{0,1} b_{0,2}) (l^B)^3 \\
&+ (2 b_{1,1} b_{0,1} b_{1,0} + b_{1,1}^2 - 2 b_{2,0} b_{0,1}^2 - 4 b_{2,0} b_{0,2} + b_{1,0}^2 b_{0,2}) (l^B)^2 \\
&+ (-4 b_{2,0} b_{0,1} + 2 b_{1,0} b_{1,1} + b_{1,0}^2 b_{0,1}) l^B - 2 b_{2,0} + b_{1,0}^2.
\end{aligned}
$$

As the coefficient $(-2q_S a_3)$ of $(c^B)^3$ is negative, the function $q^A(c^A, l^B)$ is concave for all c^B greater than the largest root of the numerator of $\frac{\partial^2 q^A(c^A,l^B)}{\partial (c^A)^2}$. If we can show that there is only one root in the positive real numbers, the sign of the numerator changes at most once from positive to negative. Thus, the function $q^A(c^A, l^B)$ is either convex, concave or it first convex and then concave.

The roots of a cubic equation can be computed by Cardano's Formula for the normal form $x^3 + ax^2 + bx + c$ (cf. [Sel70]). In our case $a := a_2/a_3$, $b := a_1/a_3$, and $c := a_0/a_3$. Note that a and b are positive because a_1, a_2, and a_3 are positive. Furthermore, we introduce:

$$p := b - \frac{a^2}{3}, \quad q := c + \frac{2a^3 - 9ab}{27}, \quad D := \frac{q^2}{4} + \frac{p^3}{27}. \tag{3.21}$$

If D is positive, there is only one real root for which a formula is known. If D is negative, which can happen in our setting, p is also negative and the three real roots are

$$z_k := 2\sqrt{\tfrac{-p}{3}} \cos\left(\tfrac{1}{3} \arccos\left(\tfrac{3q}{2p}\sqrt{\tfrac{-3}{p}}\right) - (k-1)\tfrac{2\pi}{3}\right) - \tfrac{a}{3}, \quad k = 1, 2, 3.$$

We show that z_2 and z_3 are nonpositive. The range of the function arccos is $[0, \pi]$. The domain of the cosine function is then given by $1/3[0, \pi] - (k-1)2\pi/3 = [-2(k-1)/3\pi, 1/3\pi - (k-1)2\pi/3]$ according to interval arithmetic, see Section 2.1. For $k = 2$ the domain of $\cos(x)$ is $[-2/3\pi, -1/3\pi]$ and thus, its range is $[-1/2, 1/2]$ (see Figure 3.11). For z_3 the domain of $\cos(x)$ is $[-4/3\pi, -\pi]$ and the range is $[-1, -1/2]$. This implies $z_3 \leq z_2 \leq 2\sqrt{\frac{-p}{3}}\frac{1}{2} - \frac{a}{3}$

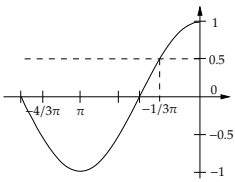


Figure 3.11.: Graph of the cosine function.

which is nonpositive as the following formula shows:

$$\sqrt{\tfrac{-p}{3}} - \tfrac{a}{3} \leq 0 \Leftrightarrow \sqrt{\tfrac{-p}{3}} \leq \tfrac{a}{3} \Leftrightarrow \tfrac{-p}{3} \leq \left(\tfrac{a}{3}\right)^2 \Leftrightarrow -\tfrac{1}{3}(b - \tfrac{a^2}{3}) \leq \tfrac{a^2}{9} \Leftrightarrow -\tfrac{1}{3}b \leq 0.$$

The first equivalence is obvious, the second follows as $p \leq 0$ and $a \geq 0$, the third from the definition of p in Equation (3.21), and the last from simple algebra. The latter expression holds since b is nonnegative. This implies that there is at most one positive root. □

An analogous argumentation can be used to derive further convex underestimators by fixing c^B to its upper bound or fixing c^A to its lower and upper bound. In a similar way concave overestimators for $q^A(c^A, c^B)$ can be determined using the lifting technique.

Computational Comparison

To compare the different relaxation strategies, we focus on a hypothetical but realistic set of parameters for the isotherms given by

$$\begin{aligned} q^A_{i,j} = q^A_{i,j}(c^A_{i,j}, c^B_{i,j}) &= \frac{5c^A_{i,j}(1 + 4c^A_{i,j} + 1c^B_{i,j})}{1 + 1c^A_{i,j} + 2c^B_{i,j} + 2(c^A_{i,j})^2 + 3(c^B_{i,j})^2 + 1c^A_{i,j}c^B_{i,j}}, \\ q^B_{i,j} = q^B_{i,j}(c^A_{i,j}, c^B_{i,j}) &= \frac{5c^B_{i,j}(2 + 6c^B_{i,j} + 1c^A_{i,j})}{1 + 1c^A_{i,j} + 2c^B_{i,j} + 2(c^A_{i,j})^2 + 3(c^B_{i,j})^2 + 1c^A_{i,j}c^B_{i,j}}. \end{aligned} \tag{3.22}$$

The relaxation strategies are applied to these isotherms and compared for the four test instances specified in Table 3.6. Test instance T1 can be seen as reference instance while T2, T3, and T4 are designed to analyze the influence of the purity requirements pur^k, the feed concentrations c^k_F, and the number of plates per zone M_i, respectively. The following computational results are not specific to the chosen parameters and domains but can be observed in general.

We construct a linear relaxation for each test instance based on the different relaxation strategies. One can verify numerically that the assumptions of Proposition 3.26 are fulfilled so that the lifting technique in RS4 can be applied. The linear relaxations are solved using CPLEX 9.10 [IBM12] in a Scheme [KCR98] framework by Utz-Uwe Haus and Matthias Köppe on a SUN FireV440 with 1.28 GHz-UltraSPARC-IIIi processors and 16 GB RAM.

$(m_1, m_2, m_3, m_4) \in [15,16] \times [3,5] \times [8,10] \times [0.25,1.25]$					
	pur^k	c_F^k	M_i	$c_{i,j}^A$	$c_{i,j}^B$
T1	0.995	0.1	30	[0.0, 0.2]	[0.0, 0.1]
T2	**0.950**	0.1	30	[0.0, 0.2]	[0.0, 0.1]
T3	0.995	**0.2**	30	[0.0, 0.4]	[0.0, 0.2]
T4	0.995	0.1	**60**	[0.0, 0.2]	[0.0, 0.1]

Table 3.6.: Parameters and domains for our test instances.

The objective in the preliminary tests is to maximize the system's throughput $m_F := m_3 - m_2$. Natural bounds on this objective function are given by the domains of the variables so that $3 \leq m_3 - m_2 \leq 7$ (see Table 3.6). In Table 3.7 the computational results are displayed which indicate that RS4 generates the tightest relaxations. RS4 proves infeasibility for T1, T3, and T4, and computes a bound for T2 which is about 50% better than the bounds of the other relaxation strategies. Interestingly, the strength of RS4 is not a result of a larger problem formulation in terms of variables and constraints, but rather it provides always the smallest description. For instance, RS4 uses only one third of the number of variables and one half of the number of constraints in the reduced LP after preprocessing as RS1. The smaller problem size is one explanation for the fast computations of RS4. It is at least twice as fast as the other relaxations, especially for the larger problem T4.

Among the relaxation strategies RS1, RS2, and RS3, which are based on a reformulation of the isotherms into structures for which the envelopes are known, RS1 computes a slightly better bound than RS2 and RS3. Recall that RS1 reformulates the isotherms into sums of bi- and trilinear terms while RS2 and RS3 exploit larger structures, namely bivariate quadratic terms. This results in a larger problem size of the relaxations in RS1. For instance, the number of variables in RS1 is the double of the one in RS2 and RS3. The larger problem sizes may explain why it needs twice the time to solve the LPs corresponding to RS1 compared to RS2 and RS3.

The faster computations of RS2 and RS3 suggest to consider refinements of a given domain with the objective to derive better bounds than RS1 in less time. For instance, consider a subdivision of the m_2-domain of $[3,5]$ into $[3,4]$ and $[4,5]$ in T4. Table 3.8 shows the bounds obtained

		RS1	RS2	RS3	RS4
T1	bound	5.64	5.77	5.90	inf.
	time	11.97	5.05	5.76	1.58
	#var/#row	2364/6195	1123/4231	1123/4476	750/3168
T2	bound	6.06	6.10	6.14	4.15
	time	11.89	6.90	5.55	3.00
	#var/#row	2364/6404	1124/4347	1124/4565	751/3716
T3	bound	6.25	6.32	6.38	inf.
	time	12.47	3.73	3.53	1.61
	#var/#row	2364/6363	1123/4235	1123/4480	751/2418
T4	bound	5.64	5.77	5.90	inf.
	time	48.25	18.95	22.18	2.70
	#var/#row	4644/12195	2203/8341	2203/8864	1470/6258

Table 3.7.: Computational results of the relaxation strategies for the objective function $\max(m_3 - m_2)$ in terms of the upper bound, the CPU time, and the number of variables and constraints in the reduced LPs.

and the computational time with and without the subdivision. With the subdivision, RS2 and RS3 are able to compute better bounds than RS1 without the subdivision in less time. The computations indicate that RS2 is slightly better than RS3 in terms of both the bound and the computational time.

All in all, the computations show that RS4 is an appropriate relaxation strategy for second-order isotherms. In contrast to the other relaxation strategies, RS4 relaxes the second-order isotherms directly and thus, its concept is closest to that of convex and concave envelopes. This shows that the knowledge of envelopes can have a significant impact on computations and that the relaxation quality of the reformulation approach can be poor. This motivates further research in the area of convex envelopes. Among the relaxation strategies RS1, RS2, and RS3, which are based on a reformulation of the isotherms, RS2 is a good trade-off between relaxation quality and computational time.

		RS1	RS2	RS3	RS4
without subdivision	bound	5.64	5.77	5.90	inf.
	time	48.25	18.95	22.18	2.70
with subdivision	bound	5.38	5.48	5.56	inf.
	time	81.36	20.94	27.10	3.63

Table 3.8.: Upper bounds on T4 and the CPU time to solve the relaxations with and without a subdivision of the m_2 domain.

3.3.3. Computing Separation Regions

Relaxation strategies RS2 and RS4 are now applied to investigate the shapes of the separation regions of continuous counter-current chromatographic processes with second order isotherms in a computational case study. Initially, we present our test set. Then, the computational methods are summarized and finally we analyze the behavior of the separation regions w.r.t. three design parameters.

Test set

We consider three series of symmetric 4-zone TMB test instances with second-order isotherms as given in Equation (3.22) in order to analyze the separation region as a function in three variables, namely the number of plates M_i per zone, the purity requirements pur^k, and the feed concentrations c_F^k:

- Test series TS1 is based on a TMB process with a relative small number of plates per zone ($M_i = 50$) and rather high purity requirements ($\text{pur}^k = 0.995$).
- Test series TS2 focuses on the same purity requirements ($\text{pur}^k = 0.995$), but the unit is equipped with a higher number of plates per zone ($M_i = 100$).
- Test series TS3 uses a TMB unit again with 100 plates per zone ($M_i = 100$), but it is characterized by lower purity requirements ($\text{pur}^k = 0.900$).

In order to incorporate the third control parameter, the feed concentration, each test series consists of eleven instances that are associated with

different feed concentrations. The specifications of the test series are summarized in Table 3.9.

	M_i	pur^k	Feed concentrations c_F^k
TS1	50	0.995	$\left\{ \begin{array}{l} 0.001, 0.01, 0.10, 0.15, \\ 0.20, 0.25, 0.30, 0.40, \\ 0.50, 1.00, 1.50 \end{array} \right\}$
TS2	100	0.995	
TS3	100	0.900	

Table 3.9.: Specifications for the three test series, where the number of plates M_i per zone, the purity requirement pur^k, and the feed concentration c_F^k are varied.

The goal of our computational study is to investigate the behavior of the separation regions in the (m_2, m_3)-space w.r.t. different process specifications. This requires to fix the flow-rate ratios m_1 and m_4 to reasonable values. The fixings and domains of the variables are given in Table 3.10. From an engineering point of view these values are chosen such that complete regeneration of the liquid phase in zones 1 and 4 is achieved and that the domains lead to efficient chromatographic systems.

Variable	m_1	m_2	m_3	m_4	$c_{i,j}^A$	$c_{i,j}^B$	c_D^k
Domain	15	[4,12]	[4,12]	1.25	$[0, 2c_F^A]$	$[0, c_F^B]$	0

Table 3.10.: Realistic domains and fixings of the variables in the TMB model.

We proved computationally that the monotonicity assumptions in Proposition 3.26 are satisfied for the isotherms in Equation (3.22) over the domain

$$0 \leq c_{i,j}^A \leq 0.7 \quad \text{and} \quad 0 \leq c_{i,j}^B \leq 0.5, \tag{3.23}$$

for all i, j, and for each fixing of the variables $c_{i,j}^A$ and $c_{i,j}^B$ to one of their bounds. As specified in Table 3.10, $c_{i,j}^A$ and $c_{i,j}^B$ are bounded from above by $u_{i,j}^A = 2c_F^A$ and $u_{i,j}^B = c_F^B$, respectively. Monotonicity is therefore guaranteed for $c_F^A = c_F^B \leq \min\{\frac{1}{2}0.7, 0.5\} = 0.35$ according to Equation (3.23). Thus,

we can employ the lifting technique of RS4 if $c_F^k \leq 0.35$, which is the case in 7 of the 11 instances for each test series in Table 3.9. For larger feed concentrations c_F^k we apply RS2 to construct the linear relaxations.

Computational Methods

We approximate the shape of the feasible separation regions from two sides. On the one hand, we use a conventional scanning technique to evaluate if certain points fulfill the restrictions of the TMB model. On the other hand, we derive infeasibility certificates for subregions with the alternative relaxation approach in order to exclude the respective subregions from the possible separation region.

Computing Operating Points by a Scanning Technique We define an equidistant grid over the (m_2, m_3) domain $[4, 12]^2$ with grid length 0.05. Using $m_2 \leq m_3$, such a grid consists of approximately 13.000 points. Each grid point is tested for feasibility by checking whether there are feasible solutions for $q_{i,j}^k$ and $c_{i,j}^k$ satisfying the mass balance equations (3.14), (3.15), (3.16), (3.17), the isotherms as given in Equation (3.22) and the purity requirements (3.18). The software SNOPT 7.2 (cf. [GMS06]) is used for this procedure.

Proving Infeasibility Regions via Linear Relaxations The (m_2, m_3) domain $[4, 12]^2$ is divided into subdomains and for each subdomain a linear relaxation is constructed and infeasibility is checked using CPLEX 9.0 [ILO07]. If the linear relaxation is infeasible, we can conclude that the original TMB model is infeasible over this subdomain. If the relaxation is feasible over a subdomain, this subdomain is divided into two smaller parts over which a new linear relaxation is constructed. The smallest size of subdomains considered throughout our computations is 0.1×0.1.

Improving the Relaxations by Bound Tightening In Chapter 2 we introduced a bound tightening technique for material balance equations whose general form is given in Equation (2.1) by

$$L_Y \, y_{i,l+1} - L_X \, x_{i,l} = L_Y \, y_{i,l} - L_X \, x_{i,l-1}, \qquad l = 2, \ldots, N-1.$$

The mass balance equations (3.15) of the TMB model are also of this form with $L_Y = 1$, $y_{i,l} = q_{i,j}^k$, $L_X = m_i$, and $x_{i,l} = c_{i,j}^k$. Analogously to the

distillation process in Chapter 2, the mass balance equations of the TMB process imply tighter bounds on the $c^k_{i,j}$-variables than the initial bounds in Table 3.10.

The idea is again to exploit tight bounds on the variables corresponding to the outlet ports, where the liquid-phase concentration variables $c^k_{2,0}$ and $c^k_{4,0}$, $k \in \{A, B\}$, have to satisfy the purity requirements (3.18). In order to illustrate our procedure, we present a bound tightening of the $c^k_{3,j}$-variables of zone 3. Improved bounds on the liquid-phase variables $c^k_{2,j}$ of zone 2 can be derived by an analogous procedure. Consider the mass balance equations (3.16) (R) and (3.15) starting at the raffinate port and going to the feed port, i.e., from the beginning of zone 4 back to the beginning of zone 3:

$$\begin{aligned} q^k_{4,1} - m_4 c^k_{4,0} - (m_3 - m_4) c^k_{4,0} &= q^k_{4,0} - m_3 c^k_{3,M_3}, \qquad (R), \\ q^k_{4,0} - m_3 c^k_{3,M_3} &= q^k_{3,j} - m_3 c^k_{3,j-1}, \qquad j = M_3, \ldots, 1. \end{aligned}$$

Let $\text{Input}^k_{4,3} := q^k_{4,1} - m_4 c^k_{4,0} - (m_3 - m_4) c^k_{4,0}$ denote the input from zone 4 to zone 3. Thus,

$$\begin{aligned} c^k_{3,M_3} &= \tfrac{1}{m_3}\Big(q^k_{4,0}(c^A_{4,0}, c^B_{4,0}) - \text{Input}^k_{4,3}\Big), \\ c^k_{3,j-1} &= \tfrac{1}{m_3}\Big(q^k_{3,j}(c^A_{3,j}, c^B_{3,j}) - \text{Input}^k_{4,3}\Big), \quad j = M_3 - 1, \ldots, 1. \end{aligned} \tag{3.24}$$

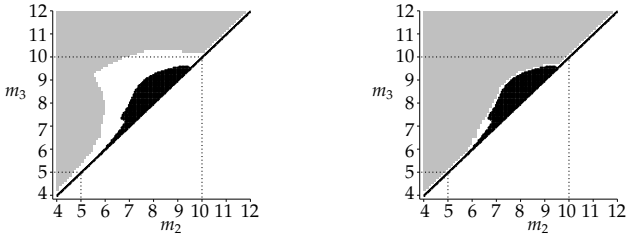
Similar to Proposition 2.5, we exploit that $q^A_{3,j}(c^A_{3,j}, c^B_{3,j})$ is nondecreasing in $c^A_{3,j}$ and nonincreasing in $c^B_{3,j}$, and $q^B_{3,j}(c^A_{3,j}, c^B_{3,j})$ is nondecreasing in $c^B_{3,j}$ and nonincreasing in $c^A_{3,j}$ to deduce bounds on the variables $c^k_{3,j}$, $j = 0, \ldots, M_i$. This, for instance, yields the following bounds for c^k_{3,M_3}

$$\begin{aligned} l^A_{3,M_3} &= \tfrac{1}{u_{m_3}}(q^A(l^A_{4,0}, u^B_{4,0}) - u^A_{\text{Input}_{4,3}}), & u^A_{3,M_3} &= \tfrac{1}{l_{m_3}}(q^A(u^A_{4,0}, l^B_{4,0}) - l^A_{\text{Input}_{4,3}}), \\ l^B_{3,M_3} &= \tfrac{1}{u_{m_3}}(q^B(u^A_{4,0}, l^B_{4,0}) - u^B_{\text{Input}_{4,3}}), & u^B_{3,M_3} &= \tfrac{1}{l_{m_3}}(q^B(l^A_{4,0}, u^B_{4,0}) - l^B_{\text{Input}_{4,3}}). \end{aligned}$$

Appropriate bounds on m_3 are obtained from the respective subdivision of the (m_2, m_3) domain. Implied bounds on $c^k_{4,0}$ and $\text{Input}^k_{4,3}$ are obtained by minimizing and maximizing the variables over the current linear relaxation.

Figure 3.12 shows the impact of the presented bound tightening technique used within RS4 on the instance from test series TS2 with feed

concentration $c_F^k = 0.25$. The gray colored area, for which infeasibility is proved, increases significantly if bound tightening is enabled.



(a) Bound tightening disabled.

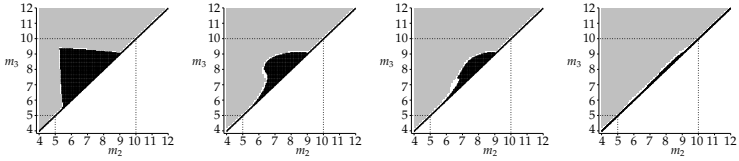
(b) Bound tightening enabled.

Figure 3.12.: The figures show the feasible operating points (black) and the infeasible regions (gray) for test series TS2 with $c_F^k = 0.25$.

Computational Results

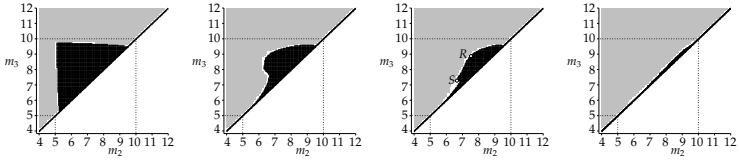
Figures 3.13–3.15 display the computed separation regions for the underlying processes with second-order isotherms. The black colored separation region is obtained by the scanning technique while infeasibility for the gray colored region is proved by the relaxation method. Initially, we show the results obtained for TMB processes and briefly discuss interesting phenomena of the separation regions regarding second-order isotherms. Then, the two approaches used to compute the separation regions are compared. Finally, we show an example which illustrates the potential of the proposed relaxation strategies in global optimization.

Phenomena Regarding Second-Order Isotherms The regions of the black colored operating points in Figures 3.13 – 3.15 depict the influence of different number of plates, different purity requirements, and different selected feed concentrations. The following trends can be observed. First, an increase in the number of plates per zone from 50 to 100 leads to an increase in the size of the separation region. Second, a decrease in the purity requirements from 99.5% to 90% increases the separation region, as well. Third, the smaller the feed concentration is chosen, the larger and the more triangular the separation region becomes. For the very small



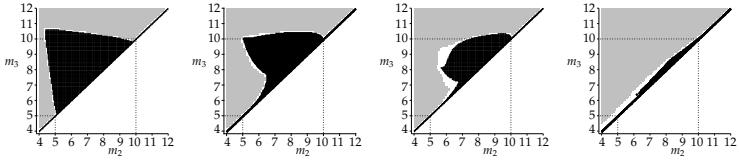
(a) $c_F^k = 0.001$ (b) $c_F^k = 0.150$ (c) $c_F^k = 0.250$ (d) $c_F^k = 1.500$

Figure 3.13.: **Feas. points** and infeas. regions for $M_i = 50$, $\text{pur}^k = 0.995$.



(a) $c_F^k = 0.001$ (b) $c_F^k = 0.150$ (c) $c_F^k = 0.250$ (d) $c_F^k = 1.500$

Figure 3.14.: **Feas. points** and infeas. regions for $M_i = 100$, $\text{pur}^k = 0.995$.



(a) $c_F^k = 0.001$ (b) $c_F^k = 0.150$ (c) $c_F^k = 0.250$ (d) $c_F^k = 1.500$

Figure 3.15.: **Feas. points** and infeas. regions for $M_i = 100$, $\text{pur}^k = 0.900$.

feed concentration $c_F^k = 0.001$ all separation regions are similar to the ideal case for linear isotherms and infinite plate numbers (cf. Figure 3.10 (a)). For a feed concentration $c_F^k = 1.500$ the separation region almost vanishes. All these effects are consistent with results of previous investigations [Maz06, SMMC93].

However, the shapes of the separation regions derived in our analysis differ significantly from the well-known shapes obtained for linear and Langmuir isotherms (cf. Figure 3.10): The separation regions for second-order isotherms are characterized by severe deviations from a triangular

shape. In particular, the maximal feed stream m_F, i.e., the maximal difference between m_3 and m_2, is not necessarily located at the *turning point* of the separation region as for processes with linear and Langmuir isotherms [Maz06]. For instance, in Figure 3.14 (c) the best known maximal feed stream over the underlying grid is obtained at the point $R := (7.35, 8.75)$, i.e., m'_F=1.40, while the turning point is given by $S := (6.65, 7.35)$ with m'_F=0.70. The distinct location of the attractive operating point R, offering twice as much feed flow, is directly related to the shapes of the adsorption isotherms and a consequence of the inflection points. Its identification with the conventional theoretical concepts used for linear and Langmuir isotherms is not possible.

The presented analysis of the separation regions hints at the practical relevance of second-order isotherms. They reflect the general trends of chromatographic processes and are capable to describe certain phenomena, like the distinct location of the turning point and the maximal feed stream, which cannot be observed for other isotherm models. More details regarding the process engineering discussion can be found in [BMSMW10].

Scanning Technique vs. Infeasibility Regions The computational results obtained by the scanning technique (black colored region) and the relaxation technique (gray colored region) lead to the same effects regarding the shape of separation regions, even though there is a gap for some instances. If high purities $\text{pur}^k = 0.995$ are requested, Figures 3.13 and 3.14 show that there is no significant gap between the computational results of the two approaches. For reduced purity requirements $\text{pur}^k = 0.9$ investigated with TS3, Figure 3.15 displays a considerable gap between the computational results of the two approaches for $c_F^k = 0.25$, $k = A, B$. These gaps are due to weaker linear relaxations for the underlying TMB models that are caused by the following two reasons: As a large feed concentration leads to larger $c_{i,j}^k$-domains, valid estimators for the involved nonlinearities must be constructed over a larger domain yielding a weaker relaxation. Moreover, the bound tightening technique gains a lot from very high purity restrictions as they imply stronger bounds on the $c_{2,0}^k$- and $c_{4,0}^k$-variables.

However, Figures 3.13 – 3.15 provide evidence that the relaxation approach is very exact for small feed concentrations that correspond to small domains. Hence, in case of higher feed concentrations further subdivi-

sions of the domains would strengthen the relaxation.

The computational results demonstrate the general applicability of the relaxation method to identify the separation regions. In contrast to the scanning technique, the relaxation method derives statements which are not only valid for discrete but for all points within a subdomain. The scanning technique may fail to reflect all details of the separation regions due to sparse scanning grids or, in a more dynamic scanning, wrong search direction or step lengths of the search direction. The relaxation method offers the possibility to overcome this risk. It gives clear certificates about the borders between feasible and infeasible operating points. This is of value, in particular, for difficult border lines occurring in case of complicated isotherm shapes (as studied in our work).

Relaxation and Global Optimization To conclude this section, the potential of the proposed relaxation techniques in the field of global optimization of chromatographic processes is briefly indicated. For the sake of illustration let us consider the instance of TS2 with feed concentration $c_F^k = 1.5$ and the objective to maximize the feed stream $m_F = m_3 - m_2$, which is an important performance indicator of chromatographic processes. The best solution found by the scanning technique over the predefined grid is $m_F' = 0.20$ (cf. Figure 3.14 (d)). The local optimization solver CoinBonmin 0.1 [BBC+08] computes a local optimum of $m_F' = 0.23$.

To evaluate the quality of the local solution, upper bounds for the maximization problem are required. The global optimization software BARON 7.8.1 [TS04] computes an upper bound of 7.99 after 62 hours on a 3GHz Dual-Core AMD Opteron(tm) Processor 8222 SE with 64 GB RAM. After seven days, the upper bound computed by BARON is still at 6.05. The trivial bound on the objective to maximize $m_F = m_3 - m_2$ over the (m_2, m_3)-domain $[4, 12]^2$ is given by 8.00 which shows that the bounds derived are of low quality. Using the ad-hoc implementation of the relaxation technique, an upper bound of 2 for m_F can be derived within 1:45 hours. After 24 hours the upper bound computed is 0.40 (cf. Figure 3.14 (d)).

Finally, note that the relaxation techniques presented can be easily adopted to other interesting optimization issues like to determine maximal productivity or maximal yield. Thus, the presented relaxation techniques can be very promising to determine optimal designs for chromatographic processes.

In this chapter we presented several techniques to derive strong convex under- and concave overestimators and illustrated their positive impact on computations. The cut-generation algorithm in Section 3.2 exploits the structural properties of bivariate functions with a fixed convexity behavior to compute cuts based on the convex envelope of the functions. The improved relaxation quality leads to a significant acceleration of the computations for instances containing functions, which are convex in one variable and concave in the other. For second-order isotherms satisfying certain monotonicity assumptions we provide estimators based on the lifting technique in Section 3.3 which clearly outperform the commonly used reformulation approach. Using the derived estimators, we analyzed chromatographic processes based on the advanced concept of second-order isotherms and revealed important chemical phenomena.

CHAPTER 4

Extended Formulations for Convex Envelopes

In the previous chapter the concept of convex envelopes of a function as best possible convex underestimator was introduced. The review of existing closed-form expressions of convex envelopes in Section 3.1 revealed that there are only a few classes of well-structured functions for which the envelopes are explicitly known. This reflects the hardness of the optimization problem corresponding to the convex envelope.

In this chapter we present convex underestimators for certain classes of functions based on a simultaneous convexification with multilinear monomials. In fact, additional constraints and variables corresponding to the monomials are added to the optimization problem which allows us to solve the problem explicitly. With this, extended formulations for the convex envelope are obtained which are as tight as the convex envelopes. The following classes of continuous functions are investigated:

Class 1: Component-wise concave (edge-concave) functions $f : [l, u] \subseteq \mathbf{R}^n \to \mathbf{R}$, $x \mapsto f(x)$.

Class 2: Functions $f : [l^x, u^x] \times [l^y, u^y] \subseteq \mathbf{R}^{n_x} \times \mathbf{R}^{n_y} \to \mathbf{R}$, $(x, y) \mapsto f(x, y)$, $n_y = 1$, which are component-wise concave in x and **convex or concave** in y whenever x is fixed to one of the vertices of the box $[l^x, u^x]$.

Class 3: Functions $f : [l^x, u^x] \times [l^y, u^y] \subseteq \mathbf{R}^{n_x} \times \mathbf{R}^{n_y} \to \mathbf{R}$, $(x, y) \mapsto f(x, y)$,

$n_y \in \mathbf{N}_{\geq 1}$, which are component-wise concave in x and **convex** in y whenever x is fixed to one of the vertices of the box $[l^x, u^x]$.

All three classes have theoretical and practical importance which motivates their further analysis. Class 1 is actually a subclass of Class 2, but due to its extensive study in the literature (cf. Subsection 3.1.1) and its possible application in the relaxation of polynomial programs (see Subsection 4.2.1), we investigate this class separately. The functions of Class 1 exhibit vertex polyhedral convex envelopes so that the determination of the convex envelope is equivalent to the analysis of the triangulations of the box $[l, u]$ (see Subsection 3.1.1 and [Tar08]). However, already the cube in dimension four exhibits 92,487,256 triangulations which can be partitioned into 247,451 symmetry classes [Pou13, HSYY08]. This is why closed-form expressions for the convex envelope of Class 1 are only known up to dimension three [MF05]. Only in the special case of submodular functions explicit formulas are known for arbitrary dimensions [TRX12].

Functions of Classes 2 and 3 are frequently used to model applications in engineering and science, which among other instances are collected in the problem libraries GLOBALLib [GLO] and MINLPLib [BDM03]. In fact, Khajavirad and Sahinidis [KS12a] revealed that up to 45% of all the nonlinear functions in these benchmark libraries are products of component-wise concave and nonnegative convex functions. Examples are functions like xy^2 over $[-1, 1]^2$ (Class 2) and $\sqrt{x}/(y_1 y_2)$ over $[1, 2]^3$ (Class 3). Considering the frequent occurrence of these functions, the knowledge about strong convex underestimators for them facilitates the computations of real world problems.

Several results are known for subclasses. Tawarmalani and Sahinidis [TS01] deduced the convex envelope for x/y (cf. Subsection 3.1.2) and generalized their idea to functions $f(x, y) : \mathbf{R} \times \mathbf{R} \to \mathbf{R}$, i.e., $n_x = n_y = 1$. Yet, their approach does not provide the convex envelope of functions f but an equivalent disjunctive programming representation.

Recently, Khajavirad and Sahinidis [KS12b, KS12a] provided the convex envelope of functions f which can be represented as $f(x, y) = g(x)h(y)$ and satisfy specific assumptions (see Subsection 3.1.3 for details). For example, g is component-wise concave and submodular, and h is a univariate convex function. This subclass of functions represents about 30% of all nonlinear terms appearing in GLOBALLib and MINLPLib. Nevertheless, their methods cannot be used to determine the convex envelope

of functions like $x_1x_2y^2$ as $g(x) = x_1x_2$ is supermodular, see Section 4.1 in [KS12a]. This function belongs to Class 2 and an extended formulation for its convex envelope is derived in this chapter.

Our approach The discussion of the different classes shows that the determination of the corresponding convex envelopes is generally hard. The envelopes are either known for low dimensional cases (Class 1) or when further assumptions on the functions are made (Classes 2 and 3). This reflects the obstacles embedded in the analytical solution of the optimization problem related to the convex envelope.

In this chapter we derive an alternative description for the convex envelope of functions belonging to Classes 1, 2, and 3 by a simultaneous convexification with multilinear monomials. For instance, let $f : [l, u] \subseteq \mathbf{R}^n \to \mathbf{R}$ be a continuous function of Class 1. We associate f with a new variable $\mu \in \mathbf{R}$ and introduce additional variables z_J for all monomials $\prod_{j\in J} x_j$ with $J \subseteq \{1, \dots, n\}, J \neq \emptyset$. The goal is to describe the following set

$$\mathcal{U}_f \quad := \quad \text{conv}(\{(z, \mu) \in \mathbf{R}^{2^n} \mid \mu \geq f(x),\ z_J = \prod_{j\in J} x_j\ \forall \emptyset \neq J \subseteq \{1, \dots, n\},\ x \in [l, u]\}),$$

where $z_J = x_j$ for all $j \in \{1, \dots, n\}$ and $J = \{j\}$.

It can be verified that the projection of $\mathcal{U}_f$ onto the (x, μ)-space corresponds to the convex envelope $\text{vex}_{[l,u]}[f]$, i.e., $\text{proj}_{(x,\mu)}(\mathcal{U}_f) = \{(x, \mu) \in \mathbf{R}^{n+1} \mid \mu \geq \text{vex}_{[l,u]}[f](x),\ x \in [l, u]\}$. Therefore, $\mathcal{U}_f$ can be interpreted as an extended formulation of $\text{vex}_{[l,u]}[f]$. On the one hand, this has the disadvantage of introducing additional variables. On the other hand, the suggested approach allows us to exploit the *Reformulation Linearization Technique* [SA90, SA94, AS05] in order to deduce closed-form expressions for the convex underestimation of f. Furthermore, this underestimation is obtained by a simultaneous relaxation of f and the multilinear monomials which can be much stronger than the individual relaxation of the functions by convex and concave envelope. Thus, the proposed relaxations do not only provide underestimators for f, but they are also of interest if both f and the monomials appear in a problem formulation.

This chapter is structured as follows. In Section 4.1 we review the Reformulation Linearization Technique. In Section 4.2 Class 1 is analyzed and the results are used to generate reduced relaxations for polynomial programs based on the Reformulation Linearization Technique. In Section 4.3 extended formulations for the convex envelopes of Classes 2 and

3 are presented. Finally, we give some computational evidence for the effectiveness of the proposed relaxations for Class 1 in Section 4.4. This chapter is based on [BM].

4.1. The Reformulation Linearization Technique

We show in the following section that the closed-form description of the extended space underestimators

$$\mathcal{U}_f \;=\; \operatorname{conv}(\{(z,\mu) \in \mathbf{R}^{2^n} \mid \mu \geq f(x),\; z_J = \prod_{j\in J} x_j \;\forall \emptyset \neq J \subseteq \{1,\dots,n\},\; x \in [l,u]\})$$

is based on a description of the convex hull of all multilinear monomials, i.e.,

$$\mathcal{S}^{(n)}_{[l,u]} \;:=\; \operatorname{conv}\left(\left\{z \in \mathbf{R}^{2^n-1} \mid z_J = \prod_{j\in J} x_j \;\forall \emptyset \neq J \subseteq \{1,\dots,n\},\; x \in [l,u]\right\}\right). \tag{4.1}$$

For this, we review the Reformulation Linearization Technique (RLT) in this section because one of its many implications is an explicit description of $\mathcal{S}^{(n)}_{[l,u]}$.

The RLT was introduced by Sherali and Adams for 0-1 linear programs [SA90]. The concept was then continuously developed further for mixed-integer 0-1 linear programs [SA94], mixed-integer linear programs [AS05], and mixed-integer semi-infinite linear and convex programs [SA09]. We will briefly summarize some of the aforementioned papers for which we also refer to the overview paper by Laurent [Lau03]. We mainly follow the notation of the mentioned papers.

Foundations of the RLT The RLT was originally developed to determine the convex hull of the following set:

$$Y = \{x \in \{0,1\}^n : Ax \leq b\},$$

where $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$. Instead of adding cutting planes to the linear relaxation to cut off fractional points, the RLT lifts the object into a higher dimension and provides a compact extended description.

The key elements of this approach are the *bound-factors* $(1 - x_j)$ and x_j which form the *bound-factor products* or *polynomial factors* $F_d[J_1, J_2](x)$,

$d = 1, \dots, n$. They are defined as

$$F_d[J_1, J_2](x) := \prod_{j \in J_1} x_j \prod_{j \in J_2} (1 - x_j) \quad \text{for } J_1, J_2 \subseteq N := \{1, \dots, n\},$$

with $J_1 \cap J_2 = \emptyset$ and $|J_1 \cup J_2| = d$. Products over the empty set are defined to be unity. By definition it is clear that for all d and all valid choices of J_1, J_2 the polynomial $F_d[J_1, J_2](x)$ is nonnegative for $x \in [0, 1]$. The term $F_d[J_1, J_2](x)$ can be equivalently written as

$$F_d[J_1, J_2](x) = \sum_{J_1 \subseteq J \subseteq J_1 \cup J_2} (-1)^{|J \setminus J_1|} \prod_{j \in J} x_j. \tag{4.2}$$

Substituting each product $\prod_{j \in J} x_j$ by a new variable z_J the linearized version of $F_d[J_1, J_2](x)$ is denoted by $f_d[J_1, J_2](z) := \sum_{J_1 \subseteq J \subseteq J_1 \cup J_2} (-1)^{|J \setminus J_1|} z_J$, where $z_{\{j\}} = x_j$ for all $j \in N = \{1, \dots, n\}$. Note that we sometimes write (x, z) to emphasize the difference between the original variables x and the additional variables z for the non-univariate multilinear monomials while sometimes we use only z to describe the entire vector.

The RLT constructs a relaxation Y_d of Y for $d = 0, \dots, n$ in a two step procedure:

Step 1 (Reformulation Step): Multiply each inequality in the description of Y by each factor $F_d[J_1, J_2](x)$ and substitute each term x_j^2 by x_j. Denote by $D = \max\{d + 1, n\}$ the largest degree of the resulting polynomials. Add all constraints $F_D[J_1, J_2](x) \geq 0$.

Step 2 (Linearization Step): Substitute each product $\prod_{j \in J} x_j$ by z_J for each $J \subseteq N$ with $|J| \neq \emptyset$.

The resulting relaxation is denoted by Y_d.

Example 4.1. Let $Y = \{(x_1, x_2) \in \{0, 1\}^2 \mid x_1 + x_2 \leq 1.2\}$. For $d = 0$, $Y_0 = \{(x_1, x_2) \in [0, 1]^2 \mid x_1 + x_2 \leq 1.2\}$ is the linear programming relaxation of Y.

For $d = 1$ the four bound factor products $F_1[\{1\}, \emptyset](x)$, $F_1[\{2\}, \emptyset](x)$, $F_1[\emptyset, \{1\}](x)$, and $F_1[\emptyset, \{2\}](x)$ are multiplied with the constraint $(x_1 + x_2) \leq 1.2$ in Step 1. For instance, $F_1[\{1\}, \emptyset](x) = x_1$ and its multiplication with the constraint yields $x_1^2 + x_1 x_2 \leq 1.2 x_1$. Substituting x_1^2 by x_1 leads to $-0.2 x_1 + x_1 x_2 \leq 0$. Step 2 transforms this constraint into $-0.2 x_1 + z_{\{1,2\}} \leq 0$. The same procedure is applied for the other bound factor products. Moreover, all constraints $F_D[J_1, J_2](x) \geq 0$ with $D = 2$ are added to the program,

e.g., $F_2[\{1,2\},\emptyset](x) = x_1x_2 \geq 0$. The first order RLT relaxation is then given by $Y_1 \subseteq \mathbf{R}^3$ with

$$Y_1 = \left\{ (x,z) \;\middle|\; \begin{array}{l} -0.2x_1 + z_{\{1,2\}} \leq 0,\ -0.2x_2 + z_{\{1,2\}} \leq 0,\ z_{\{1,2\}} \geq 0, \\ 1.2x_1 + x_2 - z_{\{1,2\}} \leq 1.2,\ x_1 + 1.2x_2 - z_{\{1,2\}} \leq 1.2, \\ x_1 - z_{\{1,2\}} \geq 0,\ x_2 - z_{\{1,2\}} \geq 0,\ -x_1 - x_2 + z_{\{1,2\}} \geq -1 \end{array} \right\}.$$

For completeness we also state the second order RLT relaxation

$$Y_2 = \left\{ (x,z) \in [0,1]^3 \;\middle|\; \begin{array}{l} z_{\{1,2\}} \leq 0,\ x_1 + x_2 - z_{\{1,2\}} \leq 1,\ -x_1 + z_{\{1,2\}} \leq 0, \\ -x_2 + z_{\{1,2\}} \leq 0,\ x_1 - z_{\{1,2\}} \geq 0,\ x_2 - z_{\{1,2\}} \geq 0, \\ -x_1 - x_2 + z_{\{1,2\}} \geq -1,\ z_{\{1,2\}} \geq 0 \end{array} \right\}.$$

◇

Sherali and Adams show that Y_d, $d = 0, \ldots, n$, constitute a hierarchy of relaxations leading to the convex hull of the set Y. More precisely, for the projection of Y_d onto the x-space, denoted by $\text{proj}_x(Y_d)$, they prove the following.

Theorem 4.2 (Theorems 1 and 3 in [SA90])**.**

$$Y_0 \supseteq \text{proj}_x(Y_1) \supseteq \cdots \supseteq \text{proj}_x(Y_n) = \text{conv}(Y).$$

The following example illustrates the increasing strength of the RLT hierarchies.

Example 4.3 (Example 4.1 continued). One can easily check that $Y = \{(0,0),(1,0),(0,1)\}$. The constraints describing Y_2 imply that $z_{\{1,2\}} = 0$ and thus, Y_2 can be written as $Y_2 = \{(x,z) \in [0,1]^3 \mid z_{\{1,2\}} = 0, x_1 + x_2 \leq 1, -x_1 \leq 0, -x_2 \leq 0\}$ which equals $\text{conv}(\{(0,0,0),(1,0,0),(0,1,0)\})$. Hence, $\text{proj}_x(Y_2) = \text{conv}(Y)$. The projection of the other relaxations are $\text{proj}_x(Y_0) = Y_0 = \text{conv}(\{(0,0),(1,0),(0,1),(1,0.2),(0.2,1)\})$ and $\text{proj}_x(Y_1) = \text{conv}(\{(0,0),(1,0),(0,1),(5/9,5/9)\})$. The different RLT relaxations of Y are shown in Figure 4.1. ◇

A central argument in the proof of Theorem 4.2 is that the vertices of the set Y_n correspond to points (x,z), where $x \in \{0,1\}^n$ and $z_J = \prod_{j \in J} x_j$ for all $J \subseteq N, J \neq \emptyset$. Thus, if the constraint set in Y is empty, the set Y_n corresponds to the simultaneous convex hull $\mathcal{S}_D^{(n)}$ of the vector of all multilinear terms up to degree n, as defined in Equation (4.1) with $D = \{0,1\}^n$.

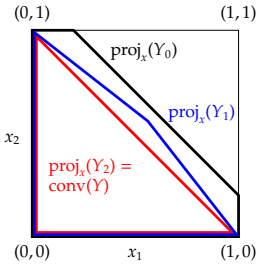


Figure 4.1.: Hierarchy of RLT relaxations for the set Y.

Theorem 4.4 (Theorem 2 in [SA90])**.** *Let* $Y = \{0,1\}^n = \{\mathbf{0},\mathbf{1}\}$*. Then,*

$$\mathcal{S}^{(n)}_{\{\mathbf{0},\mathbf{1}\}} = Y_n = \{z \in \mathbf{R}^{2^n-1} \mid f_n[J, N \setminus J](z) \geq 0 \quad \text{for all } J \subseteq N\}.$$

Moreover, the following lemma is contained in the proof of Theorem 4.4.

Lemma 4.5. *Let* $Y = \{0,1\}^n = \{\mathbf{0},\mathbf{1}\}$*. The set* Y_n *is a simplex and can be linearly transformed to the following simplex:*

$$S_n = \{y \in \mathbf{R}^{2^n-1} \mid \sum_{\emptyset \neq J \subseteq I} y_J \leq 1, \text{ and } y_J \geq 0 \text{ for all } J \subseteq I, J \neq \emptyset\}. \tag{4.3}$$

In particular, $\mathcal{S}^{(n)}_{\{\mathbf{0},\mathbf{1}\}}$ *is a simplex.*

To prove Lemma 4.5 we follow Laurent [Lau03] and use the relation between the bound factor product constraints $f_n[J, N \setminus J](z)$ and the *Zeta matrix* $\mathcal{Z}$ which is a square $0-1$ matrix of dimension $2^n \times 2^n$ with $\mathcal{Z}_{J_1,J_2} = 1$ if and only if $J_1 \subseteq J_2$. Thus, $\mathcal{Z}$ is nonsingular and its inverse reads

$$\mathcal{Z}^{-1}_{J_1,J_2} = (-1)^{|J_2 \setminus J_1|}, \text{ if } J_1 \subseteq J_2 \quad \text{and} \quad \mathcal{Z}^{-1}_{J_1,J_2} = 0, \text{ otherwise.}$$

It holds that $f_n[J, N \setminus J](z) = \mathcal{Z}^{-1}(1, z)$ which is indicated in Equation (4.2). This notation is useful in the proof of Lemma 4.5.

Proof of Lemma 4.5. This proof follows the proof of Theorem 2 in [SA90] but uses the notation of Laurent [Lau03]. The factors $F_n[J, N \setminus J](x)$, $J \subseteq N$,

sum up to 1. The same is true for the linearized factors $f_n[J, N \setminus J](z)$ which yields $f_n[\emptyset, N](z) = 1 - \sum_{\emptyset \neq J \subseteq N} f_n[J, N \setminus J](z)$. Thus,

$$Y_n = \left\{ z \in \mathbf{R}^{2^n-1} \mid f_n[J, N \setminus J](z) \geq 0 \text{ for all } \emptyset \neq J \subseteq N, \text{ and } \sum_{\emptyset \neq J \subseteq N} f_n[J, N \setminus J] \leq 1 \right\}$$
$$= \left\{ z \in \mathbf{R}^{2^n-1} \mid (\mathcal{Z}^{-1}(1, z))_J \geq 0 \text{ for all } \emptyset \neq J \subseteq N, \text{ and } \sum_{\emptyset \neq J \subseteq N} (\mathcal{Z}^{-1}(1, z))_J \leq 1 \right\}.$$

Then, apply the affine bijective transformation $\mathcal{Z}^{-1}(1, z) = (y_\emptyset, y)$ which leads to the set $S_n = \{y \in \mathbf{R}^{2^n-1} \mid y \geq 0 \text{ and } \sum_{\emptyset \neq J \subseteq N} y_J \leq 1\}$. As S_n is a simplex which can be transformed to Y_n by a nonsingular affine transformation, the set Y_n is also a simplex, cf. Lemma 4 in [SA90]. □

Therefore, the RLT provides a description for $\mathcal{S}_D^{(n)}$ if $D = \{0, 1\}^n$. The aim of the next paragraph is to obtain a description for $D = [l, u] \subseteq \mathbf{R}^n$.

Extensions to Mixed-Integer Programs Sherali and Adams generalized their concept into various directions. In [SA94] they consider mixed-integer 0-1 linear programs and in [AS05] general mixed-discrete linear programs are studied, where a subset of variables is restricted to a finite set of discrete variables. These concepts are then shown to be also valid for programs with an infinite number of linear constraints in [SA09], e.g., convex mixed-discrete 0-1 programs. The first and second extension are summarized in this paragraph since they generalize the set $\mathcal{S}_D^{(n)}$ initially to continuous boxes $D = [0, 1]^n$ and then to arbitrary continuous boxes $D = [l, u] \subseteq \mathbf{R}^n$.

For mixed-integer 0-1 linear programs Sherali and Adams analyze the convex hull of the following set:

$$Y := \{(x, y) \in \{0, 1\}^n \times [0, 1]^m : Ax + By \leq b\}.$$

Similar to the pure 0-1 case all polynomial factors $F_d[J_1, J_2](x)$ for $d = 1, \ldots, n + m$ and $J_1, J_2 \subseteq N = \{1, \ldots, n + m\}$ with $J_1 \cap J_2 = \emptyset, |J_1 \cup J_2| = d$, can be used and for $d = n + m$ the convex hull of Y in the extended space is obtained. From this the following statement can be derived.

Corollary 4.6 (cf. [SA94])**.** *Let* $Y = [0, 1]^n = [\mathbf{0}, \mathbf{1}]$. *Then,*

$$\mathcal{S}_{[\mathbf{0},\mathbf{1}]}^{(n)} = Y_n = \left\{ z \in \mathbf{R}^{2^n-1} \mid f_n[J, N \setminus J](z) \geq 0 \quad \textit{for all } J \subseteq N \right\}.$$

The authors show, however, that it is sufficient to focus on polynomial factors based on the integer variables, i.e., $d = 1, \ldots, n$ and $N = \{1, \ldots, n\}$. To this end, Step 1 of the RLT is modified such that the constraints $F_d[J_1, J_2](x) \geq y_k F_d[J_1, J_2](x) \geq 0, k = 1, \ldots, m$, are added to the problem.

Sherali and Adams present the generalization of the RLT to mixed-discrete programs in [AS05] corresponding to the following set

$$Y := \{(x, y) \in \mathbf{R}^{n+m} \mid Ax + By \leq b, y \in [0, 1]^m, x_j \in S_j \ \forall j \in N\},$$

where $N = \{1, \ldots, n\}$ and $S_j = \{\theta_{j1}, \ldots, \theta_{jk_j}\} \subseteq \mathbf{R}^{k_j}$ are finite *discrete* sets.

Recall that one reason for the strength of the original RLT relaxation is due to the substitution of x_j^2 by x_j for all binary valued variables in the reformulation step. To obtain a similar construction for mixed-discrete programs the authors make use of *Lagrange interpolating polynomials*

$$L_{jk}(x_j) = \frac{\prod_{i \in (K_j \setminus \{k\})}(x_j - \theta_{ji})}{\prod_{i \in (K_j \setminus \{k\})}(\theta_{jk} - \theta_{ji})} \quad \forall\, k \in K_j := \{1, \ldots, k_j\},$$

which can be seen as the counterpart to the 0-1 multipliers $1 - x_j$ and x_j (cf. [AS05] and references therein). For instance, let $S_j = \{l_j, u_j\}$. Then, the Lagrange interpolating polynomials read $L_{j1}(x_j) = (x_j - u_j)/(l_j - u_j)$ and $L_{j2}(x_j) = (x_j - l_j)/(u_j - l_j)$. The polynomials are multiplied with polynomials that belong to another index j and are treated similar to the factors $F_d[J_1, J_2](x)$. Note that $L_{jk}(x_j) = 1$ if $x_j = \theta_{jk}$ and $L_{jk}(x_j) = 0$ if $x_j \in S_j \setminus \{\theta_{jk}\}$. Thus, the following relation holds for each $x_j \in S_j$

$$L_{jk}(x_j)\, x_j = L_{jk}(x_j)\theta_{jk} \quad \forall k \in K_j. \tag{4.4}$$

Equation (4.4) is the key of the RLT for mixed-discrete sets. After linearization they still enforce the linearized variables to take values of their nonlinear counterparts. To illustrate this consider the 0-1 case, i.e., $S_j = \{0, 1\}$. In this setting Equation (4.4) reads $(1 - x_j) \cdot x_j = (1 - x_j) \cdot 0$ and $(x_j) \cdot x_j = (x_j) \cdot 1$. In both cases one obtains $x_j^2 = x_j$. Thus, the polynomials encode that $x_j \in S_j$.

Using the presented ideas, the RLT is then modified for mixed-discrete sets Y as follows: (i) The Lagrange interpolating polynomials $L_{jk}(x_j)$ substitute the bound factors $(1 - x_j)$ and x_j and (ii) the identity $x_j^2 = x_j$ is

replaced by Equations (4.4). The modified RLT yields a hierarchy of relaxations which leads to the convex hull of Y.

In order to obtain a description for $\mathcal{S}^{(n)}_D$, where $D = [l,u] \subseteq \mathbf{R}^n$, we set $S_j = \{l_j, u_j\}$ for all $j \in N$. We thus redefine the bound factor products as $F_d[J_1,J_2](x) := \prod_{j\in J_1}(x_j - l_j)/(u_j - l_j)\prod_{j\in J_2}(x_j - u_j)/(l_j - u_j)$. Similar to Theorem 4.5 and Corollary 4.6 we can describe $\mathcal{S}^{(n)}_{[l,u]}$ with the corresponding linearized bound factor product constraints $f_d[J_1,J_2](z)$.

Corollary 4.7 (cf. [AS05])**.** *Let $Y = \{x \in [l,u] \subseteq \mathbf{R}^n\}$. Then,*

$$\mathcal{S}^{(n)}_{[l,u]^n} = Y_n = \{z \in \mathbf{R}^{2^n-1} \mid f_d[J, N \setminus J](z) \geq 0 \quad \textit{for all } J \subseteq N\}.$$

Corollary 4.7 completes the discussion about the convex hull of all multilinear monomials over boxes $[l,u]$ which enables us to derive an extended formulation for certain classes of functions in the subsequent sections.

4.2. Component-Wise Concave Functions

In this section we derive an extended formulation for the convex envelope of component-wise concave functions $f : [l,u] \subseteq \mathbf{R}^n \to \mathbf{R}$ based on a simultaneous convexification with the vector of all multilinear monomials which is given by

$$F^{(n)} := \Big(x_1, \ldots, x_n, x_1x_2, \ldots, x_{n-1}x_n, x_1x_2x_3, \ldots, \prod_{i=1}^{n} x_i\Big).$$

For this, we introduce for each monomial $\prod_{j\in J} x_j$, $J \subseteq N := \{1,\ldots,n\}$, $J \neq \emptyset$, a new variable z_J and associate f with a new variable μ. The simultaneous convexification of the epigraph of f with the graphs of all monomials is the following convex set

$$\mathcal{U}_f := \operatorname{conv}(\{(z,\mu) \in \mathbf{R}^{2^n} \mid \mu \geq f(x),\ z = F^{(n)}(x),\ x \in [l,u]\}).$$

By definition, $\mathcal{U}_f$ provides a convex description for the underestimation of f over $[l,u]$ whose projection onto the (x,μ)-space equals the epigraph of $\operatorname{vex}_{[l,u]}[f]$.

The facet-description of the simplex $\mathcal{S}^{(n)}_{[l,u]} = \operatorname{conv}(\{z \in \mathbf{R}^{2^n-1} \mid z = F^{(n)}(x),\ x \in [l,u]\})$, which was discussed in Section 4.1, is essential for

a description of $\mathcal{U}_f$. The following elementary lemma implies that the facets of $\mathcal{S}^{(n)}_{[l,u]}$ are also facets of $\mathcal{U}_f$.

Lemma 4.8. *Let $h, g_i : D \subseteq \mathbf{R}^n \to \mathbf{R}$, $i = 1, \dots, m$, be continuous functions over a convex, compact domain $D \subseteq \mathbf{R}^n$, and let $g : D \subseteq \mathbf{R}^n \to \mathbf{R}^m$ be given by $g(x) := (g_1(x), \dots, g_m(x))^\top$. Furthermore, consider the two convex sets $\mathcal{S} := \operatorname{conv}(\{(x,z) \in \mathbf{R}^{n+m} \mid z = g(x),\ x \in D\})$ and $\mathcal{U} := \operatorname{conv}(\{(x,z,\mu) \in \mathbf{R}^{n+m+1} \mid z = g(x),\ \mu \geq h(x),\ x \in D\})$. Then, each facet-defining inequality of $\mathcal{S}$ also induces a facet for $\mathcal{U}$.*

Proof. Let $a^\top x + b^\top z \leq \gamma$ be an arbitrary facet-defining inequality for $\mathcal{S}$ with $a \in \mathbf{R}^n, b \in \mathbf{R}^m$, and $\gamma \in \mathbf{R}$. Then, $a^\top x + b^\top z \leq \gamma$ is valid for $\mathcal{U}$. As $a^\top x + b^\top z \leq \gamma$ is facet-defining for $\mathcal{S}$, there are $n+m$ points $x^r \in D$ such that the points $\left(x^r, g(x^r)\right)$, $r = 1, \dots, n+m$, are affinely independent and each point satisfies $a^\top x + b^\top z \leq \gamma$ with equality. Now, consider the set of points $(x^r, z^r, \mu^r) := \left(x^r, g(x^r), h(x^r)\right) \in \mathcal{U}$, $r = 1, \dots, n+m$, and the point $(x^{n+m+1}, z^{n+m+1}, \mu^{n+m+1}) := \left(x^1, g(x^1), h(x^1) + 1\right)$. Then, $a^\top x^r + b^\top z^r = \gamma$ and $(x^r, z^r, \mu^r) \in \mathcal{U}$ holds for all $r = 1, \dots, n+m+1$. Furthermore, the points (x^r, z^r, μ^r), $r = 1, \dots, n+m+1$ are affinely independent since the set $\{(x^r, z^r) - (x^1, z^1) \mid r = 2, \dots, n+m+1\}$ is linearly independent. □

Thus, we can apply the results of the RLT theory by Sherali and Adams [SA90, SA94, AS05] to describe $\mathcal{S}^{(n)}_{[l,u]}$ and thus, $\mathcal{U}_f$. According to Corollary 4.7 the facets of the simplex $\mathcal{S}^{(n)}_{[l,u]}$ are given by the linearized bound-factor product constraints, i.e., by

$$f_n[J, N \setminus J](z) = \left[\prod_{i \in I} (x_i - l_i) \prod_{i \in N \setminus I} (u_i - x_i) \right]_L (z) \geq 0, \quad \text{for all } I \subseteq N, \tag{4.5}$$

where the operator $[\cdot]_L(z)$ substitutes each monomial $\prod_{j \in J} x_j$ by a new variable z_J, e.g., $[-3x_1 + 5x_1x_2]_L = -3z_{\{1\}} + 5z_{\{1,2\}}$. In expanded form, the facet-defining system in Equation (4.5) yields

$$e_v(z) \ \geq \ 0, \qquad \text{for all } v \in \operatorname{vert}([l,u]), \tag{4.6}$$

where e_v denotes the linear function $e_v : \mathbf{R}^{2^n - 1} \to \mathbf{R}$ given by

$$z \quad \mapsto \quad e_v(z) \ := \ \sum_{J \subseteq N} (-1)^{\alpha(v) + |J|} \, F^{(n)}_{N \setminus J}(v) \, z_J, \tag{4.7}$$

and $\alpha(v)$ denotes the number of components of v which attain their lower bound, i.e., $\alpha(v) := \#\{i \in N \mid v_i = l_i\}$. The individual inequalities of the systems in Equation (4.5) and (4.6), which are defined by $I \subseteq N$ and $v \in \text{vert}([l,u])$, respectively, are identical if and only if $v_i = l_i$ for all $i \in I$ and $v_i = u_i$ for all $i \notin I$.

A second ingredient for deriving a description for $\mathcal{U}_f$ is the following known lemma.

Lemma 4.9 (Corollary 6 in [TS02b])**.** *Let $f : [l,u] \subseteq \mathbf{R}^n \to \mathbf{R}$ be a continuous function. There exists a unique multilinear function $m_f : [l,u] \subseteq \mathbf{R}^n \to \mathbf{R}$ which coincides with f at each vertex of the box $[l,u]$. This multilinear function reads $m_f(x) = \sum_{J \subseteq N} a_J \prod_{j \in J} x_j$ with coefficients*

$$a_J = \frac{\sum_{v \in \text{vert}([l,u])} (-1)^{\alpha(v)+|J|} \; F^{(n)}(v)_{N \setminus J} \; f(\hat{v})}{\prod_{i \in N} (u_i - l_i)}. \tag{4.8}$$

The vector $\hat{v}$ denotes the vector opposite to v in the box, i.e., $\hat{v}_j = l_j$, if $v_j = u_j$, and $\hat{v}_j = u_j$, otherwise.

The next statement provides a necessary and sufficient condition for f such that $\mathcal{U}_f$ is a polyhedral set generated by the vertices of $[l,u]$.

Theorem 4.10. *Let $f : [l,u] \subseteq \mathbf{R}^n \to \mathbf{R}$ be a continuous function. Then,*

$$\mathcal{U}_f = \{(z,\mu) \in \mathbf{R}^{2^n} \mid z \in \mathcal{S}^{(n)}_{[l,u]},\ \mu \geq [m_f]_L(z) = \sum_{J \subseteq N} a_J z_J\} \tag{4.9}$$

with a_J according to Equation (4.8)*, if and only if $f(x) \geq m_f(x) := \sum_{J \subseteq N} a_J \prod_{j \in J} x_j$ for all $x \in [l,u]$. In particular, this condition is fulfilled for component-wise concave functions f.*

Proof. By Lemma 4.8, the facet-description of $\mathcal{S}^{(n)}_{[l,u]}$ is irredundant for $\mathcal{U}_f$. It remains to discuss the additional inequality $\mu \geq [m_f]_L(z)$. If $f = m_f$, the description for $\mathcal{U}_f$ in the theorem follows easily from the fact that m_f can be uniquely represented as a linear combination of all multilinear monomials (see Lemma 4.9). If $f(x) \geq m_f(x)$ for all $x \in [l,u]$, then $\mathcal{U}_f = \mathcal{U}_{m_f}$ as f and m_f coincide at the vertices of the box which correspond to the extreme points of the set $\mathcal{U}_f$.

To prove the converse direction, assume that there is an $\bar{x} \in [l,u]$ with

$f(\bar{x}) < m_f(\bar{x})$. Then, for $(z, \mu) = (F^{(n)}(\bar{x}), f(\bar{x})) \in \mathcal{U}_f$, the relation

$$\mu = f(\bar{x}) \quad < \quad m_f(\bar{x}) = \sum_{J \subseteq N} a_J F_J^{(n)}(\bar{x}) = \sum_{J \subseteq N} a_J z_J$$

holds. This implies that $(z, \mu) \notin \{(z, \mu) \in \mathbf{R}^{2^n} \mid z \in \mathcal{S}_{[l,u]}^{(n)}, \ \mu \geq \sum_{J \subseteq N} a_J z_J\}$. Thus, $\mathcal{U}_f$ is not given by Equation (4.9). □

Theorem 4.10 extends the work of Sherali [She97] who derived an equivalent description of $\mathcal{U}_f$ for multilinear functions f. He further mentioned that for general functions f it holds that $\mathcal{U}_f = \mathcal{U}_{m_f}$ if $f \geq m_f$ over $[l, u]$. We also refer to [Taw10] in which this relation is indicated. Surprisingly, these findings were never explicitly used to construct convex underestimators of component-wise concave functions although the computational impact of this approach is tremendous as we show in Section 4.4.

The condition $f \geq m_f$ over $[l, u]$ in Theorem 4.10 implies that f must have a vertex polyhedral convex envelope (see Section 3.1.1). In fact, we have that $\text{vex}_{[l,u]}[f] = \text{vex}_{[l,u]}[m_f]$ (cf. [TS02b]). However, the extended formulation in Theorem 4.10 is not necessarily true for general functions having a vertex polyhedral convex envelope but only in the more restrictive case of $f(x) \geq m_f(x)$ for all $x \in [l, u]$.

Example 4.11. Consider $f : \mathbf{R}^2 \to \mathbf{R}$, $x \mapsto f(x) := (x_1^3 - 2x_1)(x_2^2 - 0.5)$, over $[l, u] := [-2, 1] \times [-0.75, 0.95]$. The convex envelope of f is vertex polyhedral and reads

$$\text{vex}_{[l,u]}[f](x) \ = \ \max\left\{\tfrac{1}{80}(5x_1 - 64x_2 - 58), \ \tfrac{1}{400}(161x_1 - 80x_2 - 246)\right\}.$$

The multilinear function reads $m_f(x) = -0.425 + 0.2125x_1 - 0.4x_2 + 0.2x_1x_2$ such that, for $\bar{x} = (-0.74, -0.25)$, we have that $f(\bar{x}) \approx -0.470 < -0.44525 = m_f(\bar{x})$ so that the point

$$(\bar{x}, z_{\{1,2\}}, \mu) = (\bar{x}_1, \bar{x}_2, \bar{x}_1\bar{x}_2, f(\bar{x})) = (-0.74, -0.25, 0.185, -0.470) \ \in \ \mathcal{U}_f$$

violates the additional inequality $\mu \geq [m_f]_L(z) = -0.425 + 0.2125z_1 - 0.4z_2 + 0.2z_{\{1,2\}}$. ◇

Next, we give the explicit description of $\mathcal{U}_f$ for two classes of component-wise concave functions.

Example 4.12. For $d \in \mathbf{Z}_{\geq 0}^n$, consider the negative of a monomial $x^d :=$

$\prod_{j=1}^{n} x_i^{d_i}$ over a nonnegative box $[l,u] \subseteq \mathbf{R}^n_{\geq 0}$. Then,

$$\mathcal{U}_{-x^d} = \{(z,\mu) \in \mathbf{R}^{2^n} \mid z \in \mathcal{S}^{(n)}_{[l,u]},\ \mu \geq \sum_{J \subseteq N} a_J z_J\},$$

where for all $J \subseteq N$, the coefficient a_J is equal to

$$(-1)^{n-|J|+1} \prod_{j \in N \setminus J} l_j u_j \prod_{j \in J} \left(\sum_{r=0}^{d_j-1} l_j^{d_j-1-r} u_j^r \right) \prod_{j \in N \setminus J} \left(\sum_{r=0}^{d_j-2} l_j^{d_j-2-r} u_r^j \right).$$

◇

Example 4.13. Consider a bivariate function of the form $f(x) := a_{20}x_1^{d_1} + a_{11}x_1x_2 + a_{02}x_2^{d_2}$, where $d_i \in \mathbf{Z}_{>0}$, $a_{20}, a_{11}, a_{02} \in \mathbf{R}$. If f is component-wise concave over a box $[l,u] \subseteq \mathbf{R}^2$, the facet-description of $\mathcal{U}_f$ is given by the description of $\mathcal{S}^{(2)}_{[l,u]}$ and the additional inequality

$$\left(a_{20} \sum_{i=0}^{d_1-1} l_1^{d_1-1-i} u_1^i \right) z_{\{1\}} + \left(a_{02} \sum_{i=0}^{d_2-1} l_2^{d_2-1-i} u_2^i \right) z_{\{2\}} + a_{11}\, z_{\{1,2\}} - \mu$$

$$\leq a_{20} \sum_{i=1}^{d_1-1} l_1^{d_1-i} u_1^i + a_{02} \sum_{i=1}^{d_2-1} l_2^{d_2-i} u_2^i.$$

◇

By construction, $\mathcal{U}_f$ does not only provide an underestimator for f but also a simultaneous relaxation for f and the vector of multilinear monomials $F^{(n)}$. This leads to an improved relaxation of the epigraph of f and the graphs of $F^{(n)}$ compared to the individual relaxation of the functions.

Example 4.14. Consider the function $f(x) := x_1^3x_2$ and the domain $[l,u] := [-2,1] \times [0,1]$ over which f is not component-wise concave, but it can be checked that $f(x) \geq m_f(x)$ for all $x \in [l,u]$. The convex envelope of f is vertex polyhedral and reads $\mathrm{vex}_{[l,u]}[x_1^3x_2] = \max\{3x_1 + x_2 - 3, -8x_2\}$. The extended formulation according to Example 4.12 is given by

$$\mathcal{U}_f = \{(z,\mu) \in \mathbf{R}^{2^2} \mid z \in \mathcal{S}^{(2)}_{[l,u]},\ \mu \geq [m_f]_L(z) = -2z_{\{2\}} + 3z_{\{1,2\}}\}.$$

In this setting the simultaneous relaxation of $\mu \geq f(x) = x_1^3x_2$ and the monomial $z_{\{1,2\}} = x_1x_2$ is given by $\mathcal{U}_f$. Let $\mathcal{R}$ denote the convex set obtained by the individual relaxation of f and the multilinear monomial

x_1x_2 with their corresponding envelopes, i.e.,

$$\mathcal{R} := \left\{ (z,\mu) \in \mathbf{R}^4 \;\middle|\; \begin{array}{l} z_{\{1,2\}} \geq \operatorname{vex}_{[l,u]}[x_1x_2](z_{\{1\}}, z_{\{2\}}), \\ z_{\{1,2\}} \leq \operatorname{cave}_{[l,u]}[x_1x_2](z_{\{1\}}, z_{\{2\}}), \\ \mu \geq \operatorname{vex}_{[l,u]}[f](z_{\{1\}}, z_{\{2\}}) \end{array} \right\}.$$

To measure the qualities of $\mathcal{U}_f$ and $\mathcal{R}$ in terms of relaxation, we computed their volumes using the function `NIntegrate` in MATHEMATICA 8 [Wol08]. For this, the component μ is bounded from above by $f_{\max} = \max\{f(x) \mid x \in [l,u]\} = 6$. The volumes are then given by $\operatorname{Vol}(\mathcal{U}_f, \mu \leq 6) = 11.52$ and $\operatorname{Vol}(\mathcal{R}, \mu \leq 6) = 13.20$ yielding a gap of 14%. ◇

4.2.1. Reduced RLT Relaxations for Polynomial Programs

Sherali and Tuncbilek [ST92] applied the RLT to construct tight linear relaxations of (continuous) polynomial programs. Applications and extensions of their idea can be found in several papers, see e.g., [ST97, She98, SW01, SDD12, SDL12]. Such an RLT based relaxation can, however, lead to an explosion in the problem size for instances with many variables and a high degree. One possibility to reduce the size of RLT relaxations is given in [SDL12], where the existence of a linear subsystem is exploited. We present an alternative approach to reduce the size of a RLT relaxation based on Theorem 4.10. Initially, we present the ideas of the RLT based relaxation and then, illustrate the application of Theorem 4.10.

We adapt the notation in [ST92, SDD12] and consider the polynomial program

$$\min\ \phi_0(x) \quad \text{s.t.} \quad \phi_i(x) \leq 0, \quad \forall\, i = 1,\dots,m, \quad x \in [l,u] \subseteq \mathbf{R}^n_{\geq 0}, \qquad \text{(PP)}$$

where $\phi_i(x) = \sum_{t\in T_i} \alpha_{it} \prod_{j\in J_{it}} x_j$, for $i = 0,\dots,m$. The index set T_i, $i = 0,\dots,m$, indicates the monomials occurring in $\phi_i(x)$. Let δ denote the largest degree of a monomial occurring in (PP), and let $N := \{1,\dots,n\}$ be the index set of variables. By $\overline{N}$ we denote the multiset which consists of δ copies of N, i.e., $\overline{N} = \{N, N, \dots, N\}$. Then, $J_{it} \subseteq \overline{N}$ and $|J_{it}| \leq \delta$ for all $t \in T_i$ and $i = 0, 1, \dots, m$. For instance, the multiset $\{1,1,2\}$ corresponds to the monomial $x_1^2x_2$. The classical RLT relaxation of (PP) reads

$$\begin{array}{ll} \min & [\phi_0(x)]_L(z,w) \\ \text{s.t.} & [\phi_i(x)]_L(z,w) \leq 0,\ \forall\, i = 1,\dots,m, \quad (z,w) \in R_{\text{RLT}}, \end{array} \qquad (\text{PP}_{\text{RLT}})$$

where the operator $[\cdot]_L(z,w)$ denotes the linearization of an expression such that all multilinear monomials defined by a multiset J are substituted by a new variable z_J, and all nonmultilinear monomials are substituted by a variable w_J. For example, $[-x_1^3x_2 + 5x_1x_2]_L(z,w) = -w_{\{1,1,1,2\}} + 5z_{\{1,2\}}$. The vector $(z,w) \in \mathbf{R}^{\binom{n+\delta}{\delta}-1}$ corresponds to all monomials $\prod_{j\in J} x_j$ with $\emptyset \neq J \subseteq \overline{N}$ and $|J| \leq \delta$. The set $R_{\text{RLT}} \subseteq \mathbf{R}^{\binom{n+\delta}{\delta}-1}$ is defined as

$$R_{\text{RLT}} := \Big\{(z,w) \in \mathbf{R}^{\binom{n+\delta}{\delta}-1} \mid \forall (J_1 \cup J_2) \subseteq \overline{N}, |J_1 \cup J_2| = \delta : \Big[\prod_{j\in J_1}(x_j - l_j)\prod_{j\in J_2}(u_j - x_j)\Big]_L(z,w) \geq 0\Big\}.$$

Example 4.15. Let (PP) be given as $\min\{x_1 - x_1^3 \mid x_1 \in [0,1]\}$. Then, $\delta = 3$, $\overline{N} = \{1,1,1\}$, and (PP_{RLT}) reads $\min\{z_{\{1\}} - w_{\{1,1,1\}} \mid (z,w) \in R_{\text{RLT}}\}$, where

$$R_{\text{RLT}} = \left\{ (z,w) \;\middle|\; \begin{array}{l} 1 - 3z_{\{1\}} + 3w_{\{1,1\}} - w_{\{1,1,1\}} \geq 0,\ w_{\{1,1\}} - w_{\{1,1,1\}} \geq 0 \\ z_{\{1\}} - 2w_{\{1,1\}} + w_{\{1,1,1\}} \geq 0,\ w_{\{1,1,1\}} \geq 0 \end{array} \right\}.$$

For example, the constraint $w_{\{1,1\}} - w_{\{1,1,1\}} \geq 0$ follows from $[(x_1 - 0)^2(1 - x_1)^1]_L(z,w) = [x_1^2 - x_1^3]_L(z,w) \geq 0$. Figure 4.2 illustrates the strength of the RLT relaxation. It depicts the graph of x_1^3 and the projection of the RLT relaxation to the $(z_1, w_{\{1,1,1\}})$-space.

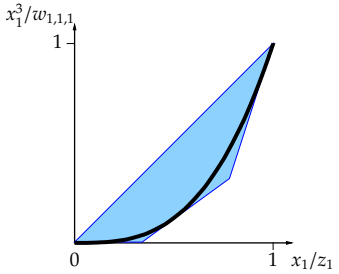


Figure 4.2.: The figure depicts the RLT relaxation (blue area) of the graph of x_1^3 (bold black line) over the interval [0,1].

◇

As already indicated and shown in the example, the RLT relaxation does not necessarily lead to the convex hull of a polynomial program (cf. Figure 4.2) but it strengthens the relaxation since the additional bound factor product constraints reflect the dependency among the different monomials. However, the size of the relaxation grows exponentially in the number of variables and the degree δ. Theorem 4.10 offers a possibility to reduce the relaxation size. If the coefficient α_{it} of a nonmultilinear monomial is negative, the corresponding term $\alpha_{it} \prod_{j \in J_{it}} x_j$ is component-wise concave and can be underestimated with the help of Theorem 4.10. We show that this term can be excluded from the determination of the largest degree δ of the program, yet yielding the same relaxation quality.

Example 4.16 (Example 4.15 continued). The component-wise concave term $(-x^3)$ is replaced by $w_{\{1,1,1\}}$. Let $f(x) := -x^3$. Theorem 4.10 yields the underestimator $-w_{\{1,1,1\}} \geq [m_f]_L(z) = -z_{\{1\}}$. Excluding the term $-x^3$, the largest degree is $\delta = 1$ and the RLT-relaxation is $z_{\{1\}} \in \mathcal{S}^{(1)}_{[0,1]} = [0,1]$. We will show that the relaxation $\min\{z_{\{1\}} - w_{\{1,1,1\}} \mid (z, w_{\{1,1,1\}}) \in R^\star_{\text{mod}}\}$ with

$$R^\star_{\text{mod}} = \left\{ \; (z, w_{\{1,1,1\}}) \;\middle|\; -z_{\{1\}} + w_{\{1,1,1\}} \leq 0, z_{\{1\}} \geq 0, z_{\{1\}} \leq 1 \; \right\}$$

is as strong as the RLT-relaxation in Example 4.15 although the relaxation based on $R^\star_{\text{mod}}$ needs one variable and one constraint less. ◇

We consider the extreme case with $\alpha_{it} < 0$ for all $t \in T_i$ and $i = 0, \ldots, m$ such that $J_{it} \not\subseteq N$, that is the coefficients of the nonmultilinear variables are negative. Recall that we consider nonnegative domains $[l, u] \subseteq \mathbf{R}^n_{\geq 0}$ so that all summands of the involved functions $\phi_i(x) = \sum_{t \in T_i} \alpha_{it} \prod_{j \in J_{it}} x_j$ are component-wise concave. We refer to this class of polynomial programs as *component-wise concave polynomial programs* (PP$^-$). Further, we assume that $\delta > n$. Otherwise, we consider the subset of monomials involved in the nonmultilinear monomial with the largest degree, e.g., for the monomials $x_1^2 x_2$, $x_1 x_2$ and, $x_1 x_2 x_3 x_4$ we just consider the monomials $x_1^2 x_2$ and $x_1 x_2$.

Technically, we proceed as follows. The index set of all nonmultilinear monomials, for which a variable w_J is introduced is denoted by $I := \{J \subseteq \overline{N} \mid 1 \leq |J| \leq \delta, \; \prod_{j \in J} x_j \text{ is nonmultilinear}\}$. The index set of nonmultilinear monomials which actually occur in (PP$^-$) is given by $I^\star := \{J_{it} \mid J_{it} \in I \text{ for } t \in T_i, \; i = 0, 1, \ldots, m\}$ and the corresponding sub-

vector of w is denoted by $w^\star$. The modified RLT relaxation reads

$$\begin{aligned} \min \quad & [\phi_0(x)]_L(z, w^\star) \\ \text{s.t.} \quad & [\phi_i(x)]_L(z, w^\star) \leq 0, \ \forall\, i = 1, \ldots, m, \quad (z, w^\star) \in R^\star_{\text{mod}}, \end{aligned} \tag{PP$^-_{\text{mod}}$}$$

where

$$R^\star_{\text{mod}} := \left\{(z, w^\star) \in \mathbf{R}^{(2^n-1)+|I^\star|} \mid z \in \mathcal{S}^{(n)}_{[l,u]}, -w^\star_J \geq [m_{-x^J}(x)]_L(z) \forall J \in I^\star\right\}$$

with $x^J := \prod_{j \in J} x_j$.

The problem characteristics of the two sets R_{RLT} and $R^\star_{\text{mod}}$ in terms of number of variables and constraints are compared in Table 4.1. Although

	R_{RLT}	$R^\star_{\text{mod}}$
#Variables	$\binom{n+\delta}{\delta} - 1$	$(2^n - 1) + \lvert I^\star \rvert$
#Constrains	$\sum_{k=0}^{\delta} \binom{n+k-1}{k} \binom{n+(\delta-k)-1}{\delta-k}$	$2^n + \lvert I^\star \rvert$
case:	$n = 5$ and $\delta = 4$	
#Variables	125	$31 + \lvert I^\star \rvert \leq 125$
#Constraints	715	$32 + \lvert I^\star \rvert \leq 126$

Table 4.1.: Problem characteristics of R_{RLT} and $R^\star_{\text{mod}}$. The formulas for R_{RLT} are from [ST92].

the problem characteristics are quite different, we prove that the two relaxations of (PP$^-$) based on R_{RLT} and $R^\star_{\text{mod}}$ return the same objective function value.

Theorem 4.17. $\min(\text{PP}^-_{\text{RLT}}) = \min(\text{PP}^-_{\text{mod}})$.

Proof. The relation $\min(\text{PP}^-_{\text{RLT}}) \geq \min(\text{PP}^-_{\text{mod}})$ can be derived as follows. Given $(\bar{z}, \bar{w}) \in R_{\text{RLT}}$, assume that its subvector $(\bar{z}, \bar{w}^\star) \notin R_{\text{mod}}$. As the RLT theory implies that $\bar{z} \in \mathcal{S}^{(n)}_{[l,u]}$ (see [ST92]), it follows that there is a $J \in I^\star$ with $-\bar{w}^\star_J < [m_{-x^J}(x)]_L(\bar{z})$. By Theorem 4.10, there exists an $\bar{x} \in [l, u]$ with $\prod_{j \in J} \bar{x}_j < m_{x^J}(\bar{x})$. This contradicts that $-\prod_{j \in J} \bar{x}_j$ is component-wise concave over the underlying positive domain and thus, we can conclude that $(\bar{z}, \bar{w}^\star) \in R_{\text{mod}}$.

For the converse relation let $(\bar{z}, \bar{w}^\star)$ be an optimal solution of $\min(\text{PP}^-_{\text{mod}})$. We can assume that $-\bar{w}^\star_J$ is at its lower bound for all $J \in I^\star$, i.e., $-\bar{w}^\star_J =$

$\sum_{S\subseteq N} a_{J,S}\bar{z}_S$ for all $J \in I^\star$, because $-\bar{w}_J^\star$ is not bounded from below by the constraints $[\phi_i(x)]_L(z, w^\star) \leq 0$, $i = 1, \ldots, m$, and the minimization of the objective function $[\phi_0(x)]_L(z, w^\star)$ attains its optimal solution at the minimal $-\bar{w}_J^\star$ (if $-\bar{w}_J^\star$ occurs in the objective function). To construct a solution $(\bar{z}, \bar{w}) \in R_{\text{RLT}}$, we define $-\bar{w}_J := \sum_{S\subseteq N} a_{J,S}\bar{z}_S$ for all $J \in I\setminus I^\star$. As $\bar{z} \in \mathcal{S}^{(n)}_{[l,u]}$, it can be represented as $\bar{z} = \sum_{v\in V} \lambda_v F^{(n)}(v)$, where $V := \text{vert}([l, u])$. Let $G : \mathbf{R}^n \to \mathbf{R}^{|I|}$, with $G_J(x) := \prod_{j\in J} x_j$ for all $J \in I$, be the vector of nonmultilinear monomials. Then, $-\bar{w}_J = \sum_{S\subseteq N} a_{J,S}(\sum_{v\in V} \lambda_v F_S^{(n)}(v)) = \sum_{v\in V} \lambda_v \sum_{S\subseteq N} a_{J,S} F_S^{(n)}(v) = \sum_{v\in V} \lambda_v m_{-xJ}(v) = \sum_{v\in V} \lambda_v(-\prod_{j\in J} v_j) = \sum_{v\in V} \lambda_v(-G_J(v))$ for all $J \in I$. Therefore, the point $(\bar{z}, \bar{w})$ can be represented as convex combination of points $(F^{(n)}(v), G(v)) \in R_{\text{RLT}}$ which shows that $(\bar{z}, \bar{w}) \in R_{\text{RLT}}$. □

One can even show that the quality of the relaxations of (PP$^-$) based on R_{RLT} and $R^\star_{\text{mod}}$ is not only identical but best possible when dealing with polynomial programs and using a relaxation which is based on the substitution of monomials by new variables. The desired object in this context is given by the convex hull of all monomials with degree less or equal to δ and reads

$$C := \text{conv}(\{(z, w) \mid z = F^{(n)}(x),\ w_J = \prod_{j\in J} x_j\ \forall J \in I,\ x \in [l, u]\}).$$

The description of C is not polyhedral and also not known in general. Let (PP^-_C) denote the relaxation of component-wise concave polynomial programs (PP$^-$) based on C. We can prove the following statement using the same arguments as in the proof of Theorem 4.17.

Theorem 4.18. $\min(\text{PP}^-_C) = \min(\text{PP}^-_{\text{RLT}}) = \min(\text{PP}^-_{\text{mod}})$.

The strength of the RLT based relaxation for (PP$^-$) provides a possible explanation for an observations made by Sherali, Dalkiran, and Desai in [SDD12] for polynomial programs: The more the programs are of the form (PP$^-$), i.e., the more negative coefficients occur, the faster the computations. The authors generated random instances which are dense and sometimes dominated by the objective function, e.g., the place in the program files occupied by the objective function varies from 15% to 90%. In particular, all monomials occur in the objective function while their occurrence in the constraints is determined randomly. The random instances are solved by the classical RLT approach and additionally by a combined approach of RLT and linear cuts derived from semidefinite

programming (SDP). Table 4.2 displays their results and shows that a higher percentage of negative coefficients in the objective function leads to a tremendous acceleration of the computations. One reason for this acceleration is given by the tight RLT based relaxation in this case.

	CPU time [s] (depending on % of neg. obj. coef.)		
	10%	50%	90%
RLT	1,173	1,850	138
RLT+SDP	674	1,055	45

Table 4.2.: The table presents the average CPU time of an RLT and a combined RLT+SDP based algorithm depending on the percentage of negative objective function coefficients. The numbers are taken from Table 3 in [SDD12].

4.3. Functions of Class 2 and 3

This section presents closed-form expressions for the convex underestimation of nonlinear functions belonging to Classes 2 and 3. Initially, we investigate Class 2 which contains continuous functions $f : [l^x, u^x] \times [l^y, u^y] \subseteq \mathbf{R}^{n_x} \times \mathbf{R} \to \mathbf{R}$, $(x, y) \mapsto f(x, y)$, that are (i) component-wise concave in the x-variables and (ii) either convex or concave in y for every fixed $\bar{x} \in \text{vert}([l^x, u^x])$. The convex envelope for a well-structured subclass of such functions was recently derived in [KS12a]. We deduce an extended formulation for the convex underestimation of the entire class of functions. This is approached by considering the following set

$$\mathcal{U}_f := \text{conv}(\{(z, \mu) \in \mathbf{R}^{2^n} \mid \mu \geq f(x, y),\ z = F^{(n)}(x, y),\ (x, y) \in [l, u]\}),$$

where $n := n_x + 1$ and $[l, u] := [l^x, u^x] \times [l^y, u^y]$.

Similar to the case of component-wise concave functions in the previous section, we show that the set $\mathcal{U}_f$ is described by the facets of $\mathcal{S}^{(n)}_{[l,u]}$ and one additional inequality which links the function $f(x, y)$ to the set of all multilinear monomials in the x- and y-variables. In contrast to the component-wise concave case, this inequality is *nonlinear* and is obtained by means of the next lemma.

Lemma 4.19. *Let* $[l,u] := [l^x,u^x] \times [l^y,u^y] \subseteq \mathbf{R}^{n_x} \times \mathbf{R}$ *be a full-dimensional box,* $N_x := \{1,\dots,n_x\}$, $V_x := \mathrm{vert}([l^x,u^x])$, *and* $n := n_x + 1$. *Moreover, let* $V_1, V_2 \subseteq V_x$ *be a partition of* V_x, *i.e.,* $V_x = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$. *For a given* $z \in \mathbf{R}^{2^n-1}$, *consider the following nonlinear system in the variables* λ_v, y^v *with* $v \in V_1$, *and* $\lambda_{v,l}$, $\lambda_{v,u}$ *with* $v \in V_2$:

$$z_J = \sum_{v\in V_1} \lambda_v F_J^{(n_x)}(v) + \sum_{v\in V_2} \Big(\lambda_{v,l} F_J^{(n_x)}(v) + \lambda_{v,u} F_J^{(n_x)}(v)\Big), \tag{4.10}$$

$$z_{J\cup\{n\}} = \sum_{v\in V_1} \lambda_v y^v F_J^{(n_x)}(v) + \sum_{v\in V_2} \Big(\lambda_{v,l} l^y F_J^{(n_x)}(v) + \lambda_{v,u} u^y F_J^{(n_x)}(v)\Big), \tag{4.11}$$

for all $J \subseteq N_x$. *Its solution is given by*

$$\lambda_v = \frac{e_{\hat{v}}(z^x)}{\prod_{j=1}^{n_x}(u_j - l_j)}, \quad y^v = \frac{\sum_{J\subseteq N_x} (-1)^{|J|+\alpha(\hat{v})}\, F_{N_x\setminus J}^{(n_x)}(\hat{v}) z_{J\cup\{n\}}}{e_{\hat{v}}(z^x)}, \tag{4.12}$$

for $v \in V_1$, *where* $\hat{v}$ *denotes the vector opposite to* v *in* $[l^x,u^x]$, z^x *denotes the subvector of* z*-variables with entries* z_J, $\emptyset \neq J \subseteq N_x$, *and* $e_{\hat{v}}(z^x)$ *according to Equation* (4.7). *The solution of* $\lambda_{v,l}$ *and* $\lambda_{v,u}$ *with* $v \in V_2$ *reads*

$$\lambda_{v,l} = \frac{e_{(\hat{v},u^y)}(z)}{\prod_{j=1}^{n}(u_j - l_j)} \quad \text{and} \quad \lambda_{v,u} = \frac{e_{(\hat{v},l^y)}(z)}{\prod_{j=1}^{n}(u_j - l_j)}. \tag{4.13}$$

Proof. We prove Lemma 4.19 in two steps. Initially, we consider subsystem (I) defined by Equation (4.10) and subsystem (II) defined by Equation (4.11) for all $J \subseteq N_x$ independently. Afterwards we combine the solutions of the two subsystems.

Let T be the matrix whose columns are given by the vectors $(1, F^{(n_x)}(v))$, $v \in V_x$. We can then bring both subsystems into the form $\zeta = T\xi$. This system has the unique solution $\xi_v = e_{\hat{v}}(\zeta)/\prod_{j=1}^{n_x}(u_j - l_j)$, $v \in V_x$ (see [SA90, Lau03, AS05]).

For subsystem (I) we replace $\Big(\lambda_{v,l} F_J^{(n_x)}(v) + \lambda_{v,u} F_J^{(n_x)}(v)\Big)$ by $(\lambda_v F_J^{(n_x)}(v))$ in Equation (4.10). Hence, we obtain the system $(1, z^x) = T\lambda$ with unique solution $\lambda_v = e_{\hat{v}}(z^x)/\prod_{j=1}^{n_x}(u_j - l_j)$, $v \in V_x$.

For subsystem (II) we substitute the term $\Big(\lambda_{v,l} l_n F_J^{(n_x)}(v) + \lambda_{v,u} u_n F_J^{(n_x)}(v)\Big)$ by $(\lambda_v y^v F_J^{(n_x)}(v))$ in Equation (4.11) and afterwards, $\lambda_v y^v$ by r_v. With $\zeta_J = z_{J\cup\{n\}}$ for all $J \subseteq N_x$, subsystem (II) is of the form $\zeta = Tr$ with unique solution

$r_v = e_{\hat{v}}(\zeta)/\prod_{j=1}^{n_x}(u_j - l_j)$, $v \in V_x$.

Finally, we consider the original system, where $r_v = \lambda_v y^v$, $v \in V_1$, and $r_v = \lambda_{v,l} l^y + \lambda_{v,u} u^y$, $\lambda_v = \lambda_{v,l} + \lambda_{v,u}$, $v \in V_2$. To obtain y^v, $v \in V_1$, we can solve $r_v = \lambda_v y^v$ for y^v if $\lambda_v \neq 0$. Then, $y^v = r_v/\lambda_v = e_{\hat{v}}(\zeta)/e_{\hat{v}}(z^x)$. If $\lambda_v = 0$, y^v can take any value as its corresponding summand cancels out.

To derive $\lambda_{v,l}$ and $\lambda_{v,u}$, $v \in V_2$, we solve the linear system $\lambda_v y^v = \lambda_{v,l} l^y + \lambda_{v,u} u^y$ and $\lambda_v = \lambda_{v,l} + \lambda_{v,u}$. Then, $\lambda_{v,l} = \lambda_v(u^y - y^v)/(u^y - l^y)$ and $\lambda_{v,u} = \lambda_v(y^v - l^y)/(u^y - l^y)$.

We prove the formula for $\lambda_{v,l}$ in Equation (4.13). An analogous argumentation holds for $\lambda_{v,u}$. We get $\lambda_{v,l} = \lambda_v(u^y - y^v)/(u^y - l^y) = e_{\hat{v}}(z^x)(u^y - y^v)/\prod_{j\in N}(u_i - l_i)$. To deduce Equation (4.13), it is thus sufficient to show that $e_{(\hat{v},u^y)}(z) = e_{\hat{v}}(z^x)(u^y - y^v)$. This follows because $e_{(\hat{v},u^y)}(z)$ can be rewritten as

$$
\begin{aligned}
&\sum_{J\subseteq N} (-1)^{|J|+\alpha(\hat{v})} F^{(n)}_{N\setminus J}(\hat{v}, u^y) z_J \\
&= \sum_{J\subseteq N_x} (-1)^{|J|+\alpha(\hat{v})} F^{(n)}_{N\setminus J}(\hat{v}, u^y) z_J + \sum_{J=T\cup\{n\}:T\subseteq N_x} (-1)^{|J|+\alpha(\hat{v})} F^{(n)}_{N\setminus J}(\hat{v}, u^y) z_J \\
&= \sum_{J\subseteq N_x} (-1)^{|J|+\alpha(\hat{v})} u^y F^{(n_x)}_{N_x\setminus J}(\hat{v}) z_J + \sum_{J\subseteq N_x} (-1)^{|J|+\alpha(\hat{v})+1} F^{(n_x)}_{N_x\setminus J}(\hat{v}) z_{J\cup\{n\}} \\
&= u^y e_{\hat{v}}(z^x) - y^v e_{\hat{v}}(z^x) = e_{\hat{v}}(z^x)(u^y - y^v).
\end{aligned}
$$

This concludes the proof. □

In the special case of $V_1 = \emptyset$, Lemma 4.19 follows from the fact that the convex hull of $\{(F^{n_x+1}(v, y),\ (v, y) \in \mathrm{vert}([l^x, u^x]\times[l^y, u^y])\}$ equals $\mathcal{S}^{(n_x+1)}_{[l^x,u^x]\times[l^y,u^y]}$. For $V_2 = \emptyset$ the solution of the corresponding system in Equations (4.12) and (4.13) is already reported in [SA94, AS05] and used to derive the equivalent extended linear formulation for certain polynomial mixed-discrete programs.

Theorem 4.20. *Consider a function $f : [l^x, u^x] \times [l^y, u^y] \subseteq \mathbf{R}^{n_x} \times \mathbf{R} \to \mathbf{R}$, $(x, y) \mapsto f(x, y)$. Let $V_x := \mathrm{vert}([l^x, u^x])$, and let $n := n_x + 1$. Assume that $f(x, y)$ is component-wise concave in x, and that V_x can be partitioned into V_1 and V_2 such that $f(x, y)$ is convex but not linear in y for each $x \in V_1$ and concave in y for each $x \in V_2$. Then,*

$$
\mathcal{U}_f = \left\{(z, \mu) \in \mathbf{R}^{2^n} \mid z \in \mathcal{S}^{(n)}_{[l,u]} \text{ and } \mu \geq \phi(z)\right\},
$$

where $\phi(z) := \sum_{v\in V_1} \lambda_v f(v, y^v) + \sum_{v\in V_2} \lambda_{v,l} f(v, l^y) + \lambda_{v,u} f(v, u^y)$, *with* λ_v *and* y^v *for* $v \in V_1$, *and* $\lambda_{v,l}$ *and* $\lambda_{v,u}$ *for* $v \in V_2$ *according to Lemma 4.19.*

Proof. Lemma 4.8 implies that the description of $\mathcal{S}^{(n)}_{[l,u]}$ is necessary for an explicit characterization of $\mathcal{U}_f$. For the remaining constraint we can argue as follows. As f is component-wise concave in the x-variables and the multilinear monomials $\prod_{j\in J} x_j$ are linear in the x-variables, the set $\mathcal{U}_f$ can be represented as (see [TS02b, Taw10]):

$$\mathcal{U}_f \;=\; \operatorname{conv}\Big(\bigcup_{v\in V_x} \Big\{ (F^{(n)}(v, y^v), \mu) \mid \mu \geq f(v, y^v),\; y^v \in [l^y, u^y] \Big\} \Big).$$

For each fixed $v \in V_x$ the set $\mathcal{U}_{f(v,y)}$ corresponds to the epigraph of the function $\operatorname{vex}_{[l^y,u^y]}[f_v]$, where $f_v(y) := f(v, y)$. If $v \in V_1$, then $f(v, y)$ is convex and $\operatorname{vex}_{[l^y,u^y]}[f_v](y) = f(v, y)$. If $v \in V_2$, then $f(v, y)$ is concave and $\operatorname{vex}_{[l^y,u^y]}[f_v](y)$ is given by the secant connecting $(l^y, f_v(l^y))$ and $(u^y, f_v(u^y))$.

Disjunctive programming techniques imply that, for any given $\bar{z} \in \mathcal{S}^{(n)}_{[l,u]}$, the corresponding minimal value μ with $(\bar{z}, \mu) \in \mathcal{U}_f$ can be computed by the following optimization problem

$$\begin{aligned}
\min \quad & \sum_{v\in V_1} \lambda_v f(v, y^v) && + \sum_{v\in V_2} \left(\lambda_{v,l} f(v, l^y) + \lambda_{v,u} f(v, u^y)\right) \\
\text{s.t.} \quad \bar{z} = \; & \sum_{v\in V_1} \lambda_v F^{(n)}(v, y^v) && + \sum_{v\in V_2} \left(\lambda_{v,l} F^{(n)}(v, l^y) + \lambda_{v,u} F^{(n)}(v, u^y)\right) \\
1 = \; & \sum_{v\in V_1} \lambda_v && + \sum_{v\in V_2} (\lambda_{v,l} + \lambda_{v,u}) \\
& \lambda_v \geq 0,\; v \in V_1, \quad \lambda_{v,l}, \lambda_{v,u} \geq 0,\; v \in V_2, \quad y^v \in [l^y, u^y],\; v \in V_1 \;.
\end{aligned}$$

The constraint set of this problem is solved in Lemma 4.19. Note that $\lambda_v \geq 0$, $v \in V_1$, $\lambda_{v,l}, \lambda_{v,u} \geq 0$, $v \in V_2$, and $1 = \sum_{v\in V_1} \lambda_v + \sum_{v\in V_2} (\lambda_{v,l} + \lambda_{v,u})$ follows from the fact that $\bar{z} \in \mathcal{S}^{(n)}_{[l,u]}$. This proves the claim. □

Remark 4.21. Theorem 4.10 is a special case of Theorem 4.20, namely for $V_1 = \emptyset$ and $V_2 = V_x$. Even though the two representations do not coincide at a first glance, it can be checked that the additional inequality in Theorem 4.20 reduces to the one in Theorem 4.10 in this special case.

The next example illustrates Theorem 4.20 and emphasizes its potential for simultaneous convexification purposes.

Example 4.22. Let $f(x, y) = x_1 x_2 / y$, $x_1 \in [-1, 1], x_2 \in [0.1, 1], y \in [0.1, 1]$. This is Example 2 in [KS12a]. The convex envelope over the subdomain

$0.9x_1 + 2x_2 \geq 1.1$ reads

$$\mathrm{vex}_{[l,u]}[f](x,y) = \begin{cases} \frac{(0.5x_1+1.1x_2-0.6)^2}{y+0.05x_1+0.11x_2-0.16} + 5x_1 - 1.1x_2 - 3.9, & \text{if } 0.1 \leq y \leq s_1, \\ \frac{(0.5x_1+0.76x_2-0.26)^2}{y+0.05x_1-0.05} + 5x_1 - 5, & \text{if } s_1 \leq y \leq s_2, \\ \frac{0.12(x_2-1)^2}{y-0.45x_1-1.1x_2+0.56} + 5.5x_1 + 1.1x_2 - 5.6, & \text{if } s_2 \leq y \leq s_3, \\ 10y + x_1 + x_2 - 11, & \text{if } s_3 \leq y \leq 1, \end{cases}$$

where $s_1 = 10y+x_1+x_2-11$, $s_2 = 0.45x_1+0.76x_2-0.21$, and $s_3 = 0.45x_1+0.55$. The convex envelope over $0.9x_1 + 2x_2 \leq 1.1$ reads

$$\mathrm{vex}_{[l,u]}[f](x,y) = \begin{cases} \frac{0.5(x_1+1)^2}{20y+x_1-1} + 0.5x_1 - 10x_2 + 0.5, & \text{if } 0.1 \leq y \leq s_3, \\ y + 0.1x_1 - 10x_2, & \text{if } s_3 \leq y \leq 1.1 - x_2, \\ 10y + 0.1x_1 - x_2 - 9.9, & \text{if } 1.1 - x_2 \leq y \leq 1. \end{cases}$$

The extended formulation $\mathcal{U}_f$ is given by the facets of $\mathcal{S}^{(3)}_{[l,u]}$ and the inequality $\mu \geq \phi(z)$ with

$$\begin{aligned} \phi(z) := &- 5.5z_{\{2\}} + 5.5z_{\{1,2\}} + 5z_{\{2,3\}} - 5z_{\{1,2,3\}} - \tfrac{101}{81} \\ &+ \tfrac{(1+z_{\{1\}}-z_{\{2\}}-z_{\{1,2\}})^2}{18(z_{\{3\}}+z_{\{1,3\}}-z_{\{2,3\}}-z_{\{1,2,3\}})} + \tfrac{(1+z_{\{1\}}-10z_{\{2\}}-10z_{\{1,2\}})^2}{180(-z_{\{3\}}-z_{\{1,3\}}+10z_{\{2,3\}}+10z_{\{1,2,3\}})}. \end{aligned}$$

The set $\mathcal{U}_f$ is the simultaneous convex hull of (z,μ) with $\mu \geq f(x,y)$ and the seven multilinear monomials in the x- and y-variables over $[l,u]$. Let $\mathcal{R}$ denote the convex relaxation, where f and each multilinear monomial is individually relaxed by its convex and concave envelope (cf. Example 4.14). We can bound component μ from above by $\max\{f(x) \mid (x,y) \in [l,u]\} = 10$. The volumes of the individually and simultaneously convexified sets computed with MATHEMATICA 8 [Wol08] are $\mathrm{Vol}(\mathcal{R}, \mu \leq 10) \approx 0.325$ and $\mathrm{Vol}(\mathcal{U}_f, \mu \leq 10) \approx 0.014$. This yields a gap of 2120%. ◇

Remark 4.23. Note that it is not clear how to generalize Theorem 4.20 to functions, where the convex part consists of more than one component, i.e., functions $f : [l^x, u^x] \times [l^y, u^y] \subseteq \mathbf{R}^{n_x} \times \mathbf{R}^{n_y}$, $(x,y) \mapsto f(x,y)$, with $n_y > 1$. For instance, let $n_x = 1$ and $n_y = 2$ with $V_1 = \{l^x\}$ and $V_2 = \{u^x\}$. Following the proof of Theorem 4.20, the inequality

$$\mu \geq \lambda_l f(l^x, y^l) + \sum_{s \in \mathrm{vert}([l^y, u^y])} \lambda_{u,s} f(u^s, s)$$

is part of the description for $\mathcal{U}_f$. This inequality involves seven unknowns λ_l, y_1^l, y_2^l and $\lambda_{u,s}$ with $s \in \text{vert}([l^y, u^y])$ and $|\text{vert}([l^y, u^y])| = 4$. In the proof of Theorem 4.20 we further exploit the fact that for a fixed $\bar{x}$ the set $\mathcal{U}_{f(\bar{x},y)}$ corresponds to the epigraph of the function $\text{vex}_{[l^y,u^y]}[f_{\bar{x}}]$, where $f_{\bar{x}}(y) := f(\bar{x}, y)$. This is not the case for $n_y = 2$ since the monomial y_1y_2 needs to be taken into account for $\mathcal{U}_{f(\bar{x},y)}$, as well. To overcome this hurdle, one might only introduce the monomials x_1y_1 and x_1y_2. However, this leads only to a system of six linear equations, namely for $z_{\{1\}}, z_{\{2\}}, z_{\{3\}}, z_{\{1,2\}}, z_{\{1,3\}}$, and the summation condition for the convex multipliers. As there are seven variables, it is still necessary to solve an optimization problem for the remaining unknown variable in order to determine $\mathcal{U}_f$.

Next, we consider Class 3 which contains continuous functions $f : [l^x, u^x] \times [l^y, u^y] \subseteq \mathbf{R}^{n_x} \times \mathbf{R}^{n_y} \to \mathbf{R}$, $(x, y) \mapsto f(x, y)$, that are component-wise concave in the x-variables and convex on the space of the y-variables for every fixed $x \in \text{vert}([l^x, u^x])$. Let $N_x := \{1, \dots, n_x\}$ and $N_y := \{1, \dots, n_y\}$. We introduce

- for all $J \subseteq N_x$, $J \neq \emptyset$, the monomials $\prod_{j\in J} x_j$ and the variables $z_J \in \mathbf{R}$,

- for all $k \in N_y$ and for all $J \subseteq N_x$, the monomials $y_k \prod_{j\in J} x_j$ and the variables $w_J^k \in \mathbf{R}$, where we define $w^k := (w_\emptyset^k, w_{\{1\}}^k, \dots, w_{N_x}^k)$ which is associated with

$$y_k(1, F^{(n)}(x)) \quad = \quad (y_k, y_kx_1, \dots, y_kx_n, y_kx_1x_2, \dots, y_k \prod_{j=1}^{n} x_j).$$

This collection of monomials ensures that for a fixed x all introduced monomials are either constant or linear. An extended formulation for the convex envelope of f is then given by the set

$$\begin{aligned} \mathcal{E}_f \;:=\; \text{conv}\Big(\Big\{(z, w^1, \dots, w^{n_y}, \mu) \in \mathbf{R}^{(2^{n_x}-1)+n_y\cdot 2^{n_x}+1} \;\Big|\; & \mu \geq f(x, y), \\ & z = F^{(n_x)}(x),\ w^k = y_k(1, F^{(n_x)}(x)),\ \text{for all } k \in N_y, \\ & z_{\{j\}} = x_j \in [l_j^x, u_j^x],\quad j = 1, \dots, n_x, \\ & w_\emptyset^k = y_k \in [l_k^y, u_k^y],\quad k = 1, \dots, n_y\Big\}\Big). \end{aligned}$$

By construction and our assumptions on f, $\mathcal{E}_{f(v,y)}$ corresponds to the epigraph of $\text{vex}_{[l,u]}[f(v, y)] = f(v, y)$, for all $v \in \text{vert}([l^x, u^x])$. Similar to

the description of $\mathcal{U}_f$ which is based on $\mathcal{S}^{(n)}_{[l,u]}$, Lemma 4.8 implies that the description of the following set is needed for $\mathcal{E}_f$.

$$\begin{aligned}\mathcal{L}^{(n_x,n_y)}_{[l,u]} := \operatorname{conv}\Big(\Big\{(z, w^1, \dots, w^{n_y}) \in \mathbf{R}^{(2^{n_x}-1)+n_y\cdot 2^{n_x}} \;\Big|\; & z = F^{(n_x)}(x),\\ & w^k = y_k(1, F^{(n_x)}(x)), \text{ for all } k \in N_y,\\ & z_{\{j\}} = x_j \in [l^x_j, u^x_j], \quad j = 1, \dots, n_x,\\ & w^k_\emptyset = y_k \in [l^y_k, u^y_k], \quad k = 1, \dots, n_y\Big\}\Big)\end{aligned}$$

Sherali and Adams showed that the set $\mathcal{L}^{(n_x,n_y)}_{[l,u]}$ can be represented as intersection of the simplices $\mathcal{S}^{(n_x+1)}_{[l^x,u^x]\times[l^y_k,l^y_k]}$.

Lemma 4.24 ([SA94, AS05]). $\mathcal{L}^{(n_x,n_y)}_{[l,u]} = \bigcap_{k=1}^{n_y}\{(z, w^1, \dots, w^{n_y}) \mid (z, w^k) \in \mathcal{S}^{(n_x+1)}_{[l^x,u^x]\times[l^y_k,l^y_k]}\}$.

According to our definition, points in $\mathcal{S}^{(n_x+1)}_{[l^x,u^x]\times[l^y_k,l^y_k]}$ are labeled by subsets $J \subseteq \{1, \dots, n_x + 1\}$, $J \neq \emptyset$, that follow the order of the vector $F^{(n_x+1)}$. This labeling might be different to the order of the vector (z, w^k). However, to keep the notation simple, we assume for Lemma 4.24 that the components of points (z, w^k) are permuted in the correct way when necessary.

We are now ready to give a description for $\mathcal{E}_f$.

Theorem 4.25. *Let $f : [l^x, u^x] \times [l^y, u^y] \subseteq \mathbf{R}^{n_x} \times \mathbf{R}^{n_y} \to \mathbf{R}$ be a function that is component-wise concave in the x-variable for every fixed $y \in [l^y, u^y]$ and convex on the space of y-variables for every $\bar{x} \in V := \operatorname{vert}([l^x, u^x])$. Then,*

$$\mathcal{E}_f = \left\{ (z, w^1, \dots, w^k, \mu) \;\middle|\; \begin{array}{l} (z, w^k) \in \mathcal{S}^{(n_x+1)}_{[l^x,u^x]\times[l^y_k,u^y_k]}, \quad k = 1, \dots, n_y, \\ \mu \geq \varphi(z, w) := \sum_{v\in V} \lambda_v \, f(v, y^v) \end{array} \right\},$$

where, for all $v \in V$ and $k \in N_y$,

$$\lambda_v = \frac{e_{\hat{v}}(z)}{\prod_{i=1}^{n_x}(u^x_i - l^x_i)}, \quad y^v_k = \frac{\sum_{J\subseteq N_x}(-1)^{|J|+a(\hat{v})}\, F^{(n_x)}_{N_x\setminus J}(\hat{v}) w^k_J}{e_{\hat{v}}(z)}, \tag{4.14}$$

$e_{\hat{v}}(z)$ according to Equation (4.7), and $\hat{v}$ is the vector opposite to v in $[l^x, u^x]$.

Proof. The constraints $(z, w^k) \in \mathcal{S}^{(n_x+1)}_{[l^x,u^x]\times[l^y_k,u^y_k]}$, $k \in N_y$, are implied by Lemmas 4.8 and 4.24. For the remaining constraint we can argue similar to the

proof of Theorem 4.20 with $V_2 = \emptyset$. Moreover, the representation of $\mathcal{L}_{[l,u]}^{(n_x,n_y)}$ in Lemma 4.24 implies that for each $k \in \{1,\dots,n_y\}$ the linear systems corresponding to y_k^v, $v \in V$, can be solved independently (see [SA94]). Thus, for each $k \in \{1,\dots,n_y\}$ the formula for y_k^v in Equation (4.14) is given analogously to Equation (4.12), where $n_y = 1$.

□

Remark 4.26. The condition of being component-wise concave in the x-variables in Theorems 4.20 and 4.25 can be relaxed to the condition $f(x,\bar{y}) \geq m_{f(x,\bar{y})}(x)$ for all $x \in [l^x,u^x]$ and all fixed values $\bar{y} \in [l^y,u^y]$, where $m_{f(x,\bar{y})}(x)$ is the multilinear function obtained in Lemma 4.9. If we consider the special case of $f(x,y) = g(x)h(y) \neq 0$ with $h(y)$ nonnegative and convex, we can strengthen Theorems 4.20 and 4.25 as follows. The extended formulation in Theorem 4.20 is valid if and only if $g(x) \geq m_g(x)$ for all $x \in [l^x,u^x]$. If $g(x)$ is further nonnegative, the extended formulation in Theorem 4.25 is valid if and only if $g(x) \geq m_g(x)$ for all $x \in [l^x,u^x]$.

The next example illustrates Theorem 4.25 and compares the extended formulation to the convex envelope.

Example 4.27. Let $f(x,y) = x/(y_1y_2)$, $(x,y_1,y_2) \in [l,u] := [0.5,2] \times [0.1,1] \times [1.5,2]$. This is Example 2 in [KS12b], where the convex envelope of f is described by six different formulas, each of them valid over a specific subdomain of the box $[l,u]$. The extended formulation $\mathcal{E}_f$ obtained by the simultaneous convexification with the monomials $y_1x\,(= w^1_{\{1\}})$ and $y_2x\,(= w^2_{\{1\}})$ is given by

$$\mathcal{E}_f = \left\{ (z,w^1,w^2,\mu) \in \mathbf{R}^5 \;\middle|\; \begin{array}{l} (z,w^1) \in \mathcal{S}^{(2)}_{[0.5,2]\times[0.1,1]}, (z,w^2) \in \mathcal{S}^{(2)}_{[0.5,2]\times[1.5,2]}, \\ \mu \geq \varphi(z,w^1,w^2) \end{array} \right\},$$

where

$$\varphi(z,w^1,w^2) := \tfrac{l^x(u^x-z)^3}{(u^x-l^x)(u^xw^1_\emptyset - w^1_{\{1\}})(u^xw^2_\emptyset - w^2_{\{1\}})} + \tfrac{u^x(z-l^x)^3}{(u^x-l^x)(l^xw^1_\emptyset - w^1_{\{1\}})(l^xw^2_\emptyset - w^2_{\{1\}})}.$$

The variable μ can be bounded from above by $\max\{f(x,y) \mid x \in [l,u]\} = 40/3$. Mathematica 8 computes the volumes of $\mathcal{E}_f$ and its individual counterpart $\mathcal{R}$ as $\mathrm{Vol}(\mathcal{E}_f, \mu \leq 40/3) \approx 0.263$ and $\mathrm{Vol}(\mathcal{R}, \mu \leq 40/3) \approx 0.269$ which implies a gap of 2%. ◇

In Examples 4.22 and 4.27, some advantages and disadvantages of the extended formulations $\mathcal{U}_f$ and $\mathcal{E}_f$ compared to the convex envelopes are

indicated. The extended formulations have the disadvantage of introducing additional variables corresponding to certain multilinear monomials. Especially for higher dimensional functions, the exponential growth in the number of variables can lead to an explosion of the problem size. Nevertheless, for lower dimensional cases the growth of variables is reasonable and we noticed that the multilinear monomials often occur in the problem description, see e.g., problems `ex734` and `ex735` from GLOBALLib [GLO] and `eniplac`, `1252`, `nvs05`, and `pump` from MINLPLib [BDM03]. Therefore, the extended formulations can lead to improved convex relaxations as indicated in Example 4.22. Furthermore, one can check that the formulas describing parts of the convex envelope are only valid over the specified subdomains. For instance, consider Example 4.22 and let $(\bar{x}_1, \bar{x}_2, \bar{y}) = (0, 0.5, 0.7)$. Then, $\text{vex}_{[l,u]}[f](\bar{x}_1, \bar{x}_2, \bar{y}) = \bar{y} + 0.1\bar{x}_1 - 10\bar{x}_2 = -4.3$ while this is violated by the last formula, $10\bar{y} + 0.1\bar{x}_1 - \bar{x}_2 - 9.9 = -3.4$. Usually, convex relaxations are constructed and solved over the entire domain. Thus, the formulas of the convex envelope can be used in a cut-generation algorithm to construct valid linear cuts, but they cannot be added directly to the convex relaxation whereas this is possible with the extended formulation.

To conclude this section, we emphasize that the convex envelopes for the two classes of functions considered in this section are not known, in general. As discussed in Subsection 3.1.3, Khajavirad and Sahinidis [KS12b, KS12a] have recently derived explicit formulas of convex envelopes for special subclasses in the original space. They consider functions $f(x, y) = g(x)h(y)$, where

- $g(x)$ is a component-wise concave function such that its restriction to the vertices is submodular and has the same monotonicity in every argument,
- $h(y)$ is a nonnegative convex function of one of the two forms (i) $h(y) = y^a$, $a \in \mathbf{R} \setminus [0, 1]$ or (ii) $h(y) = a^y$, $a > 0$, and
- $g(x)$ is nonnegative or $h(y)$ is monotone.

For special cases they relax some conditions but the assumptions above reflect their general setting.

First, the formulations presented in this chapter do not require that f can be written as $f(x, y) = g(x)\ h(y)$. For instance, the function $f(x, y) = (y + 1)\exp(xy)$ over $[l, u] = [-1, 1] \times [-3, -1]$ belongs to the functions

considered in Theorem 4.20 but does not fit into the concept of Khajavirad and Sahinidis.

Second, Khajavirad and Sahinidis state that the property of component-wise concavity of $g(x)$ can be relaxed to having a vertex polyhedral convex envelope in their context. For the extended formulation, we can relax the component-wise concavity of $g(x)$ by $g(x) \geq m_g(x)$ for all $x \in [l^x, u^x]$ (see Remark 4.26). In this case, the assumptions of Khajavirad and Sahinidis are more general than ours. For example, the function $g(x) = \max\{-x_1 + 0.5, -x_2 + 0.5\}$ over $[l^x, u^x] = [0, 1]^2$ is vertex polyhedral, submodular when restricted to the vertices of $[l^x, u^x]$ and nonincreasing in each argument. However, $g(x) < m_g(x) = -x_1x_2 + 0.5$ for all x in the interior of the box $[0, 1]^2$.

Third, in the setting of the convex envelope the univariate variable y in the convex function $h(y)$ can be replaced by $c^T y + d$, where y is multivariate, if $g(x)$ is nonnegative. This extension is also covered by Theorem 4.25 because $f(x, c^T y + d)$ is the composition of a convex and a linear function w.r.t. to the y-space and thus, it is convex [Roc70].

Finally, Theorems 4.20 and 4.25 do not require that $g(x)$ is submodular when restricted to the vertices and nondecreasing (or nonincreasing) in every argument. For instance, the convex envelope of the function $f(x, y) = g(x)h(y) = (x_1x_2)y^2$ cannot be determined by the framework of Khajavirad and Sahinidis as g is supermodular (cf. Section 4.1 in [KS12b]) while the function satisfies all assumptions of Theorems 4.20.

4.4. Computations

In this section we present a computational case study which compares our extended formulations with standard relaxation methods. We focus on the component-wise concave functions discussed in Section 4.2 because their extended formulation is polyhedral and thus, easier to implement. Yet, the presented results can hint at the computational behavior of the extended formulations for the other two classes of functions. In the following, we first present our test set which consists of instances of the Molecular Distance Geometry Problem. Second, different relaxation strategies for this class of problems are investigated. Finally, we implemented two separators for the MINLP solver SCIP [Ach09] which are based on the relaxations $\mathcal{S}^{(n)}_{[l,u]}$ and $\mathcal{U}_f$. We apply these implementations to our test set and compare their results to the results of state-of-the-art software.

Molecular Distance Geometry Problem

The Molecular Distance Geometry Problem (MDGP) (see e.g., [LLM09]) is to determine the three-dimensional structure of a molecule consisting of a finite set $A = \{1, \dots, s\}$ of atoms and given distances $d_{\{i,j\}} \geq 0$ between two atoms $\{i, j\} \in E \subseteq A \times A$ (edge set). This leads to the following unconstrained nonconvex optimization problem

$$\min \sum_{\{i,j\} \in E} \left(\|\xi^i - \xi^j\|^2 - d^2_{\{i,j\}} \right)^2 \quad \text{s.t.} \quad \xi := (\xi^1, \dots, \xi^s) \in \mathbf{R}^{3s}, \tag{4.15}$$

where $\xi^i := (\xi^i_1, \xi^i_2, \xi^i_3) \in \mathbf{R}^3$, $i = 1, \dots, s$, represents the position of atom i in the three-dimensional space. A point $\xi \in \mathbf{R}^{3s}$ is a solution of the MDGP if and only if the corresponding objective function value is zero.

In the formulation of Equation (4.15) the MDGP can be solved instantaneously by solvers like BARON or SCIP for low dimensional problems. In order to illustrate the impact of the proposed relaxation methods, we follow [CLL10] and analyze the *expanded model* formulation

$$\min \sum_{\{i,j\} \in E} s_{\{i,j\}} \text{ s.t. } s_{\{i,j\}} \geq \text{EXPAND}\Big[\left(\|\xi^i - \xi^j\|^2 - d^2_{\{i,j\}} \right)^2 \Big], \ \xi \in \mathbf{R}^{3s}, \tag{4.16}$$

where the operator EXPAND$[\cdot]$ expands each term $\left(\|\xi^i - \xi^j\|^2 - d^2_{\{i,j\}} \right)^2$ such that it is given as the sum of 52 monomials of the following form:

$$x_1, \quad x_1x_2, \quad x_1x_2x_3, \quad x_1x_2x_3x_4, \quad x_1^2, \quad x_1^4, \quad -x_1^2x_2x_3, \quad -x_1^3x_2.$$

We consider two test sets related to the MDGP. Test set TS1 contains five MDGP instances lavor6 through lavor20 which are characterized in Table 4.3. The instances differ in the number of atoms and edges, and the domains which are chosen such that the instances are feasible. All instances (except for the domain) were randomly generated as described in [Lav06] and given to us by Jon Lee. Test set TS2 consists of 50 randomly generated test instances, where we construct 10 random instances for each of the five Lavor instances. For this, it is decided uniformly at random if a summand is multiplied by zero or one. Thus, the instances of TS2 are sparser than the instances of TS1.

Instance	lav6	lav7	lav8	lav10	lav20
#atoms/#edges	6 / 13	7 / 16	8 / 20	10 / 28	20 / 70
domain	$[0,3]$	$[0,4]$	$[0,4]$	$[0,5]$	$[0,9]$

Table 4.3.: Lavor instances: Each instance is characterized by the number of atoms, the number of edges between the atoms, and the domain of each component ξ_k^i of an atom i.

Different Relaxation Strategies Applied to TS1

We consider four different linear relaxation strategies which we briefly summarize. Relaxation strategy **StandRelax** follows Cafieri et al. [CLL10], where each term is reformulated into terms of products of univariate or bilinear/trilinear terms for which the formulas of their envelopes are applied. **QHullRelax** additionally computes the convex envelopes for all component-wise concave monomials by the algorithm Qhull [BDH96]. In $\mathcal{S}$**-Relax** all multilinear terms, in particular, the quadrilinear terms $x_1x_2x_3x_4$ are relaxed by $\mathcal{S}^{(4)}_{[l,u]}$. $\mathcal{U}$**-Relax** follows $\mathcal{S}$-Relax and further employs extended space underestimators $\mathcal{U}_f$ for the component-wise concave monomials $f(x) = -x_1^3x_2$ and $f(x) = -x_1^2x_2x_3$.

All computations were accomplished with SCIP 2.1.1 [Ach09] using the LP solver CPLEX 12.3 [IBM12] on a 2.67 GHz INTEL X5650 with 96 GB RAM. QHullRelax uses Qhull 2012.1 [BDH96]. The time limit for all computations is one hour.

All relaxation strategies are employed in a branch-and-bound framework and the results concerning the bound obtained at the root node, the final bound, and the number of iterations are displayed in Table 4.4. Note that the optimal objective function value for all instances is zero. Several observations can be made. First, the root node relaxations of $\mathcal{S}$-Relax and $\mathcal{U}$-Relax are twice as good as the relaxations of StandRelax and QHullRelax. As we start with lower bounds of zero for the variables in the root node, StandRelax and QHullRelax, and $\mathcal{S}$-Relax and $\mathcal{U}$-Relax yield the same lower bound. Changing the lower bounds to 1, for instance, reveals that QHullRelax generates stronger bounds than StandRelax and $\mathcal{U}$-Relax is better than $\mathcal{S}$-Relax.

Second, the final bounds derived by $\mathcal{S}$-Relax and $\mathcal{U}$-Relax are always twice as good as the bound obtained by StandRelax and QHullRelax. This

		StandRelax	QHullRelax	$\mathcal{S}$-Relax	$\mathcal{U}$-Relax
lav6	root	-36,871	-36,871	-14,770	-14,777
	1 hour	-15,554	-21,727	-7,212	-6,333
	#iter	14,406	750	13,182	18,147
lav7	root	-141,278	-141,278	-56,698	-56,698
	1 hour	-69,754	-99,271	-32,564	-30,002
	#iter	12,039	365	11,008	15,649
lav8	root	-176,869	-176,869	-70,946	-70,946
	1 hour	-100,891	-138,822	-46,212	-43,218
	#iter	10,090	231	8,839	12,689
lav10	root	-602,754	-602,754	-241,694	-241,694
	1 hour	-423,748	-520,950	-184,078	-176,735
	#iter	6,898	138	6,372	9,703
lav20	root	-15,840,033	-15,840,033	-6,367,589	-6,367,589
	1 hour	-13,291,564	-14,557,815	-5,618,446	-5,529,058
	#iter	2,690	44	2,192	3,360

Table 4.4.: Test set TS1: Comparison of the behavior of the relaxation strategies concerning their root node relaxation, the final bound (1 hour), and the number of iterations in the branching procedure. All computations were stopped after one hour.

shows that the extended space relaxations are not only stronger but are also solvable in a comparable time. For instance, $\mathcal{U}$-Relax performs always the highest number of iterations among all relaxation strategies. Besides the stronger relaxation quality of $\mathcal{U}$-Relax, the higher number of iterations is a reason why the final bound of $\mathcal{U}$-Relax compared to $\mathcal{S}$-Relax is about 10 % better for the smaller Lavor instances and still 3 % better for the larger instances.

It is noticeable that relaxation strategy QHullRelax returns always the worst bound. This is due to the expensive computation of the convex envelope by the Qhull algorithm. In order to analyze the possible impact of the convex envelopes, we compare the bounds of StandRelax and QHullRelax after the same number of iterations in Table 4.5. The bounds of the two relaxations are almost the same although QHullRelax employs

additional convex envelopes. An analysis of StandRelax and QHullRelax shows that most of the constraints describing the additional convex envelopes are already implied by the constraints of StandRelax: The projection of the relaxation for the component-wise concave functions used in StandRelax based on reformulation is almost identical to the relaxation by the convex envelopes and even links the different functions to each other.

	lav6	lav7	lav8	lav10	lav20
iter	750	365	231	138	44
StandRelax	-21,730	-99,271	-138,822	-520,950	-14,557,815
QHullRelax	-21,727	-99,271	-138,822	-520,950	-14,557,815

Table 4.5.: Final bounds by StandRelax and QHullRelax after the same number of iterations.

A Comparison of Standard Solvers Applied to TS2

In this subsection we compare the computational results of the state-of-the-art solver BARON [TS05], the open-source solver SCIP [Ach09], and SCIP with two separators based on the derived extended formulations. The separators are add-ons for SCIP and can be downloaded from `http://www.ifor.math.ethz.ch/staff/balmarti`. The separator SimMultMonom is based on $\mathcal{S}^{(n)}_{[l,u]}$ while the separator CWConcaveMonom uses $\mathcal{U}_f$, where f is a monomial over a nonnegative domain. We denote the corresponding algorithms $\mathcal{S}$-SCIP and $\mathcal{U}$-SCIP, respectively.

All computations were accomplished with BARON 11.1.0 and SCIP 3.0.0. We used the default settings of the separators except for the frequency which is set to 1 in order to apply the separators at every iteration. The current implementation of the separators requires to reformulate the problems such that additional variables are introduced corresponding to the monomials which are then linked to the monomials by additional constraints. We refer to this formulation as *reformulated model* formulation. Both BARON and SCIP were tested on the reformulated model and the expanded model formulation in Equation (4.16). As both algorithms perform better on the expanded model formulation, we subsequently state only their results for this formulation.

Table 4.6 shows the computational results for test set TS2 consisting of 50 randomly modified Lavor instances. We compare the algorithms in terms of four criteria: The number of times an algorithm computes the best lower or upper bound on the problem or is at most 0.01% worse than the best bound. The dual gap is computed w.r.t. the best known feasible solution over all algorithms as the arithmetic sum over all instances, where the gap for each instance is either bounded by 100% or 1000%. See the discussion of Table 3.2 in Section 3.2.3 for a detailed description of the performance criteria.

	BARON	SCIP	$\mathcal{S}$-SCIP	$\mathcal{U}$-SCIP
#best primal/dual bound	18 / 0	10 / 0	22 / 3	22 / 50
dual gap (100%)	84.65%	97.73%	58.50%	55.19%
dual gap (1000%)	140.14%	256.41%	64.44%	60.52%

Table 4.6.: Test set TS2 (50 randomized Lavor instances): Comparison of the number of times an algorithm computes the best lower or upper bound or is in the range of the best bound, and the sum of the dual gaps over all instances, where each summand is either bounded by 100% or 1000%.

Good primal bounds are computed by the algorithms BARON, $\mathcal{S}$-SCIP, and $\mathcal{U}$-SCIP. The primal bounds of all algorithms deviate in average not more than 6% from the best primal bound. The best dual bounds are obtained by $\mathcal{U}$-SCIP for all cases, but the dual gaps show that $\mathcal{S}$-SCIP is almost as good as $\mathcal{U}$-SCIP. In particular, the dual gaps by algorithms $\mathcal{S}$-SCIP and $\mathcal{U}$-SCIP are two times better than the dual gap of BARON and four times better than the dual gap of SCIP in the 1000% case. This comparison shows that SCIP can benefit from the separators and that using the separators in BARON may even yield better results.

Finally, we remark that the algorithms $\mathcal{S}$-SCIP and $\mathcal{U}$-SCIP introduce variables corresponding to the monomials needed for the relaxations $\mathcal{S}^{(n)}_{[l,u]}$ and $\mathcal{U}_f$. In contrast to test set TS1, not all of the corresponding monomials occur in the problem formulation of TS2 so that the problem formulation becomes much bigger than necessary. Nevertheless, the results show that the proposed relaxations accelerate the computations significantly.

In this chapter we suggested an alternative approach to derive closed-

form expressions for convex envelopes in an extended space. Our approach relies on the introduction of additional variables corresponding to multilinear monomials. This allows us to reduce the obstacles involved in the computation of the convex envelopes and to exploit the RLT theory to provide explicit formulas for three classes of interesting functions f. In fact, the extended formulations correspond to a simultaneous convexification of f with the multilinear monomials. These relaxations can be much stronger than the individual relaxations of the functions by their convex and concave envelopes and can also accelerate computations as shown in the last part of this chapter.

CHAPTER 5

Simultaneous Convexification

In the previous chapter the convexification of several functions simultaneously was considered as an auxiliary approach to overcome the combinatorial difficulties involved in the determination of convex envelopes. The corresponding fast computations (see Section 4.4) and the large difference in volume of the individual and the simultaneous relaxations (see Example 4.22) motivated us to explicitly study the *simultaneous convex hull* of the graph of several functions in this chapter.

Definition 5.1. Let $f : D \subseteq \mathbf{R}^n \to \mathbf{R}^m$, $x \mapsto f(x) = (f_1(x), \dots, f_m(x))$, be a vector-valued function. The convex hull $Q_D[f] := \operatorname{conv}(\{(x, z) \in \mathbf{R}^{n+m} \mid z = f(x), x \in D\})$ of the graph of f over D is called the *simultaneous convex hull* of f over D. The convex hull of the epigraph of f over D is denoted by $\mathcal{E}_D[f] := \operatorname{conv}(\{(x, z) \in \mathbf{R}^{n+m} \mid z \geq f(x), x \in D\})$.

Literature on the simultaneous convex hull of a vector of functions over *continuous* domains is rare (cf. [Taw10]). Vectors of general functions are hardly investigated while specific vectors of well-structured functions are analyzed for decades. The most prominent class is the vector of all monomials for n variables up to degree δ. One possibility to relax the simultaneous convex hull of this vector is the Reformulation Linearization Technique (RLT) presented in Section 4.1. If the subvector of all multilinear monomials is considered, the RLT even provides the simultaneous convex hull. For the general vector of monomials and $n = 1$ the moment curve $(x_1^1, x_1^2, \dots, x_1^\delta)$ is obtained whose simultaneous convex hull is known for arbitrary $\delta \in \mathbf{N}$ and $[l, u] \subseteq \mathbf{R}$ due to [KS53]. For $\delta = 2$

the vector of functions consists of all quadratic monomials, which is of particular importance for quadratically constrained quadratic programs. Its simultaneous convex hull is known up to $n = 3$ [AB10]. Note that all mentioned vectors are in some sense "complete", e.g., they contain *all* quadratic monomials formed by the variables $x_1, \dots, x_n$, so that certain structural properties can be exploited to deduce $Q_D[f]$.

Tawarmalani [Taw10] was the first to address the simultaneous convex hull $Q_D[f]$ of a vector $f(x) = (f_1(x), \dots, f_m(x))$ of general functions over continuous domains D. The author focuses on the set of extreme points of $Q_D[f]$ and provides criteria to exclude points from this set. If the set of extreme points is the disjunctive union of subsets and the simultaneous convex hull can be described restricted to these subsets, he suggests to apply disjunctive programming techniques to derive *extended formulations* for the overall simultaneous convex hull. Furthermore, it is shown that the convex hull $\mathcal{E}_D[f]$ of the epigraph of the vector f is obtained by intersecting the convex hulls of the epigraphs of the individual functions f_i.

In contrast to Tawarmalani's approach that mainly considers the simultaneous convex hull as an object on its own, we link the simultaneous convex hull to convex envelopes and exploit the rich theory of convex envelopes to derive properties of $Q_D[f]$. Initially, we extend the work of Tawarmalani regarding the extreme points of $Q_D[f]$ and show that the union of the extreme points of $Q_D[\sum_{i=1}^m \alpha_i f_i] \subseteq \mathbf{R}^{n+1}$ over all $\alpha \in \mathbf{R}^m$ is dense in the set of extreme points of $Q_D[f] \subseteq \mathbf{R}^{n+m}$ w.r.t. the x-components. Then, we focus on the generation of valid inequalities for $Q_D[f]$. Instead of using disjunctive programming as suggested by Tawarmalani, we follow an alternative approach to directly derive valid inequalities in the original space of $Q_D[f]$. The basis for this approach is our finding that the high dimensional object $Q_D[f] \subseteq \mathbf{R}^{n+m}$ for $f : D \subseteq \mathbf{R}^n \to \mathbf{R}^m$ can be represented via the intersection of the lower dimensional objects $Q_D[(\sum_{i=1}^m \alpha_i f_i)] \subseteq \mathbf{R}^{n+1}$ with $\alpha \in \mathbf{R}^m$:

$$\begin{aligned} Q_D[f] &= \bigcap_{\alpha \in \mathbf{R}^m} \{(x, z) \in \mathbf{R}^{n+m} \mid (x, \alpha^\top z) \in Q_D[(\alpha^\top f)]\} \\ &= \bigcap_{\alpha \in \mathbf{R}^m} \{(x, z) \in \mathbf{R}^{n+m} \mid \operatorname{vex}_D[\alpha^\top f](x) \leq \alpha^\top z,\ x \in D\}. \end{aligned} \tag{5.1}$$

This representation allows us to derive strong valid constraints for $Q_D[f]$ using the different methods for convex underestimation of a function

$\alpha^\top f : D \subseteq \mathbf{R}^n \to \mathbf{R}$, especially using the convex envelope (see Chapter 3).

A central question for the representation in Equation (5.1) is whether all $\alpha \in \mathbf{R}^m$ are needed to describe $\mathcal{Q}_D[f]$. For $D = [l, u] \subseteq \mathbf{R}^n$ we identify the subsets C_{vex} and $C_{\text{poly},T}$ which are **not** necessary for $\mathcal{Q}_D[f]$. The set C_{vex} is the cone of all α for which $\alpha^\top f$ is convex. For each triangulation T of $[l, u]$ we introduce a cone $C_{\text{poly},T}$, where $\alpha \in C_{\text{poly},T}$ if and only if $\text{vex}_{[l,u]}[\alpha^\top f]$ is vertex polyhedral and its polyhedral subdivision of $[l, u]$ corresponds to T. If the two types of cones are closed, the interior points can be excluded from the representation of $\mathcal{Q}_D[f]$. We explicitly determine the cones C_{vex} and $C_{\text{poly},T}$ for vectors of two and three univariate convex functions. Note that results for these "simple" functions can have a significant impact on computations since higher dimensional functions, whose convex envelope are not known, are often reformulated as sums and products of univariate functions. This reformulation often leads to subsystems consisting of (convex) univariate functions as shown in Example 1.1.

For the vector of two univariate convex functions we further prove that all $\alpha \in \mathbf{R}^2$, which are not in the interior of the cones C_{vex} and $C_{\text{poly},T}$, are irredundant in the description of $\mathcal{Q}_D[f]$ via Equation (5.1), i.e., the corresponding constraint $\text{vex}_D[\alpha^\top f](x) \leq \alpha^\top z$ cannot be obtained by a conic combination of other constraints $\text{vex}_{[l,u]}[(\alpha^i)^\top f](x) \leq (\alpha^i)^\top z$, $\alpha^i \in \mathbf{R}^2$. Besides this, a separation result is presented to cut off any given point $(x, z) \notin \mathcal{Q}_D[f]$.

This chapter is organized as follows. We start with a literature overview in Section 5.1. In Section 5.2 we relate $\mathcal{Q}_D[f]$ to convex envelopes and with this, we obtain basic properties of $\mathcal{Q}_D[f]$. In Section 5.3 vectors of two and three univariate convex functions are investigated and strong relaxations are derived. In Section 5.4 we give computational evidence for the impact of the new relaxations. The results of this chapter are joint work with Dennis Michaels and Robert Weismantel.

5.1. Overview of Simultaneous Convexification

Initially, two specific vectors f of functions are considered for which the simultaneous convex hull or at least tight relaxations are known: The moment curve $f(x) = (x^2, x^3, \ldots, x^\delta)$, $\delta \in \mathbf{N}_{\geq 2}$ and the vector of quadratic functions $f(x) = (x_1^2, \ldots, x_n^2, x_1x_2, \ldots, x_{n-1}x_n)$. Then, we present the work of Tawarmalani [Taw10] in detail, where properties of $\mathcal{Q}_D[f]$ are derived

for general vectors f.

5.1.1. The Moment Curve

The simultaneous convex hull of the vector $f(x) := (x^2, \ldots, x^\delta)$, $\delta \in \mathbf{N}_{\geq 2}$, over the domain $[l, u] = [0, 1]$ is well-studied as an auxiliary object of the *truncated Hausdorff moment problem* (cf. [KS53] and references therein): Given a finite sequence of numbers $\{\mu_1, \ldots, \mu_\delta\}$, the problem is to find a positive Borel measure μ on $[0, 1]$ such that the numbers μ_k, $k = 1, \ldots, \delta$, are the k-th moments of μ, i.e., $\mu_k = \int_0^1 t^k \, d\mu(t)$ for all $k = 1, \ldots, \delta$. Karlin and Shapeley [KS53] prove that such a measure exists if and only if $(\mu_1, \ldots, \mu_\delta) \in Q_{[0,1]}[f]$. Moreover, they provide a description of $Q_{[0,1]}[f]$ via the *Hankel determinants*

$$\underline{H}_{2k}(\mu) = \begin{vmatrix} 1 & \mu_1 & \cdots & \mu_k \\ \vdots & & & \vdots \\ \mu_k & \mu_{k+1} & \cdots & \mu_{2k} \end{vmatrix},$$

$$\overline{H}_{2k}(\mu) = \begin{vmatrix} \mu_1 - \mu_2 & \mu_2 - \mu_3 & \cdots & \mu_k - \mu_{k+1} \\ \vdots & & & \vdots \\ \mu_k - \mu_{k+1} & \mu_{k+1} - \mu_{k+2} & \cdots & \mu_{2k-1} - \mu_{2k} \end{vmatrix},$$

$$\underline{H}_{2k+1}(\mu) = \begin{vmatrix} \mu_1 & \mu_2 & \cdots & \mu_{k+1} \\ \vdots & & & \vdots \\ \mu_{k+1} & \mu_{k+2} & \cdots & \mu_{2k+1} \end{vmatrix},$$

$$\overline{H}_{2k+1}(\mu) = \begin{vmatrix} 1 - \mu_1 & \mu_1 - \mu_2 & \cdots & \mu_k - \mu_{k+1} \\ \vdots & & & \vdots \\ \mu_k - \mu_{k+1} & \mu_{k+1} - \mu_{k+2} & \cdots & \mu_{2k} - \mu_{2k+1} \end{vmatrix}.$$

Theorem 5.2 (Theorems 7.2, 7.3, 17.2, and 17.3 in [KS53])**.** *Let* $f(x) := (x^2, \ldots, x^\delta)$, $\delta \in \mathbf{N}_{\geq 2}$. *Then,*

$$Q_{[0,1]}[f] = \{z \in \mathbf{R}^\delta \mid \underline{H}_k(z) \geq 0, \overline{H}_k(z) \geq 0, k = 1, \ldots, \delta\}.$$

A point $z = (z_1, \ldots, z_\delta) \in \mathbf{R}^\delta$ *is contained in the interior of* $Q_{[0,1]}[f]$ *if and only if all* $\underline{H}_k(z) > 0$ *and* $\overline{H}_k(z) > 0$ *for all* $k = 1, \ldots, \delta$. *A point* $z \in \mathbf{R}^\delta$ *is contained in the boundary of* $Q_{[0,1]}[f]$ *if and only if there is an* r *with* $1 \leq r \leq \delta$ *such that* $\underline{H}_k(z) > 0$ *and* $\overline{H}_k(z) > 0$ *for all* $k = 1, \ldots, r - 1$, $\underline{H}_r(z) = 0$ *or* $\overline{H}_r(z) = 0$, *and*

$\underline{H}_k(z) = 0$ *and* $\overline{H}_k(z) = 0$ *for all* $k = r+1, \ldots, \delta$.

The boundary of $Q_{[0,1]}[f]$ can be described by the functions $\underline{z}_\delta(z)$, $\overline{z}_\delta(z) : [0,1]^\delta \to \mathbf{R}$ with

$$\underline{z}_\delta(z) := z_\delta - \frac{\underline{H}_\delta(z)}{\underline{H}_{\delta-2}(z)} \quad \text{and} \quad \overline{z}_\delta(z) := z_\delta + \frac{\overline{H}_\delta(z)}{\overline{H}_{\delta-2}(z)}$$

in the following way:

Theorem 5.3 (Section 18 in [KS53])**.** *Let* $f(x) := (x^2, \ldots, x^\delta)$, $\delta \in \mathbf{N}_{\geq 2}$. *The functions* $\underline{z}_\delta(z)$ *and* $\overline{z}_\delta(z)$ *are independent of* z_δ, *and* $z = (z_1, \ldots, z_\delta) \in Q_{[0,1]}[f]$ *if and only if* $\underline{z}_\delta(z) \leq z_\delta \leq \overline{z}_\delta(z)$.

Example 5.4. If $\delta = 2$, then $f(x) = (x^2)$ and

$$\underline{z}_2(z) = z_2 - \frac{z_2 - z_1^2}{1} = z_1^2 \quad \text{and} \quad \overline{z}_2(z) = z_2 + \frac{z_1 - z_2}{1} = z_1$$

such that $Q_{[0,1]}[(x^2)] = \{z \in \mathbf{R}^2 \mid z_1^2 \leq z_2 \leq z_1\}$. If $\delta = 3$, then $f(x) = (x^2, x^3)$ and

$$\underline{z}_3(z) = z_3 - \frac{z_1 z_3 - z_2^2}{z_1} = \frac{z_2^2}{z_1} \quad , \quad \overline{z}_3(z) = z_3 + \frac{(-1+z_1)z_3 + z_2 + z_1 z_2 - z_1^2 - z_2^2}{1 - z_1} = \frac{z_2 + z_1 z_2 - z_1^2 - z_2^2}{1 - z_1}$$

such that $Q_{[0,1]}[(x^2, x^3)] = \left\{ z \in \mathbf{R}^3 \mid \frac{z_2^2}{z_1} \leq z_3 \leq \frac{z_2 + z_1 z_2 - z_1^2 - z_2^2}{1 - z_1} \right\}$. ◇

In the subsequent sections we propose a relaxation for $Q_{[l,u]}[f]$ when f is a vector of two or three univariate convex functions which is similar to the linear relaxation given by Karlin and Shapely for $Q_{[l,u]}[f]$ when $f = (x^2, \ldots, x^\delta)$. They show that the convex set $Q_{[0,1]}[f]$ is included in a simplex $S^\delta \subseteq \mathbf{R}^\delta$ whose vertices $v^k = (v_1^k, \ldots, v_\delta^k), k = 0, \ldots, \delta$, are given by $v_i^k = \binom{k}{i} / \binom{\delta}{i}$, $i = 1, \ldots, \delta$. For instance, in case of $\delta = 2$ the simplex S^2 exhibits the vertices $v^0 = (0,0)$, $v^1 = (1/2, 0)$, and $v^2 = (1,1)$. The comparison of $Q_{[0,1]}[(x^2)]$ and S^2 in Figure 5.1 reveals that the points v^k are chosen such that the segments (v^0, v^1) and (v^1, v^2) correspond to the tangents on $f(x) = (x^2)$ at $x = 0$ and $x = 1$.

The presented theory can be extended to arbitrary intervals $[l, u] \subseteq \mathbf{R}$ by means of a nonsingular, affine transformation T. For this, it is exploited that $z_1' \in [l, u]$ if and only if $z_1 = \frac{z_1' - l}{u - l} \in [0,1]$. Based on this, the transformation T represents each z_i as the linearized version of $\left(\frac{z_1' - l}{u - l}\right)^i$ for

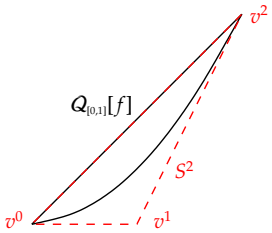


Figure 5.1.: $Q_{[0,1]}[f]$ (black) with $f(x) = (x^2)$ and S^2 (red).

all $i = 1, \ldots, \delta$, i.e.,

$$z_i = T_i(z') := \tfrac{1}{(u-l)^i} \sum_{j=0}^{i-1} (-1)^j \binom{i}{j} l^j \, z'_{i-j} + \tfrac{(-1)^i l^i}{(u-l)^i}, \quad i = 1, \ldots, \delta. \tag{5.2}$$

Then, $z \in Q_{[0,1]}[f]$ if and only $z' \in Q_{[l,u]}[f]$. Moreover, replacing z by $T(z)$ in $Q_{[0,1]}[f]$ leads to the combinatorial equivalent set $Q_{[l,u]}[f]$ (cf. [SA90]).

To conclude the discussion of moment curves, we illustrate the potential of the simultaneous convex hull $Q_{[l,u]}[f]$ compared to the standard relaxation of the functions by convex and concave envelopes and compared to the simplex S^δ.

Example 5.5. Let $\delta = 3$, $f(x) = (x^2, x^3)$, and $[l, u] = [1, 2]$. Usually the individual functions are relaxed by the functions itself as convex envelopes and by the secants of the functions as concave envelope yielding the standard relaxation

$$R_{\text{Std}} = \left\{ (z_1, z_2, z_3) \in \mathbf{R}^3 \mid z_1^2 \le z_2 \le 3z_1 - 2, \; z_1^3 \le z_3 \le 7z_1 - 6 \right\}.$$

The affine transformation in Equation (5.2) evolves to

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & -3 & 1 \end{pmatrix} z' + \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}.$$

Using this transformation, the set $Q_{[0,1]}[(x^2, x^3)]$ from Example 5.4 changes

to

$$Q_{[1,2]}[(x^2, x^3)] = \left\{(z_1, z_2, z_3) \in \mathbf{R}^3 \mid \tfrac{z_1^2 - z_2 - z_1 z_2 + z_2^2}{z_1 - 1} \leq z_3 \leq \tfrac{4z_2 + 2z_1 z_2 - 4z_1^2 - z_2^2}{2 - z_1}\right\}.$$

Furthermore, the affine transformation of the simplex S^3 by T is denoted by $S^3_{[1,2]}$ and its outer description reads

$$\begin{aligned} 12z_1 - 8 \leq \ \ 6z_2 - z_3, \qquad & 5z_1 - 2 \leq \ \ 4z_2 - z_3, \\ -3z_1 + 1 \leq -3z_2 + z_3, \qquad & -8z_1 + 4 \leq -5z_2 + z_3. \end{aligned}$$

The volumes of the three convex relaxations of the moment curve are computed using `NIntegrate` in Mathematica 8 [Wol08] and are given in Table 5.1. The standard relaxation is improved by a factor of 8 by the "simultaneous" linear relaxation $S^3_{[1,2]}$ and by a factor of 27 by the simultaneous convex hull $Q_{[1,2]}[(x^2, x^3)]$. The numbers show that the linear relaxation $S^3_{[1,2]}$ is a strong relaxation for $Q_{[1,2]}[(x^2, x^3)]$. ◇

	R_{Std}	$S^3_{[1,2]}$	$Q_{[1,2]}[(x^2, x^3)]$
Volume	0.1500	0.0185	0.0055

Table 5.1.: Volume of the different convex relaxations for $Q_{[1,2]}[(x^2, x^3)]$.

In Section 5.3 we use the moment curve as a reference to evaluate our relaxations for the vectors of univariate convex functions.

5.1.2. Quadratic Monomials

Quadratically constrained quadratic programs have been a central topic in the optimization community over the last decades (see [BST11, BS12] for an overview). Among others, a classical approach to solve these programs is to relax the vector of all possible quadratic monomials, i.e., $f : [l, u] \subseteq \mathbf{R}^n \to \mathbf{R}^{\binom{n+1}{2}}$ with $f(x) := (x_1^2, \ldots, x_n^2, x_1 x_2, \ldots, x_{n-1} x_n)$. For this, each monomial $x_i x_j$ is associated with a new variable $z_{i,j}$ for all $1 \leq i \leq j \leq n$ such that

$$Q_{[l,u]}[f] = \operatorname{conv}\left\{(x, z) \in \mathbf{R}^{n + \binom{n+1}{2}} \mid z_{i,j} = x_i x_j \text{ for all } 1 \leq i \leq j \leq n,\ x \in [l, u]\right\}.$$

To keep notation short, we consider the case of $[l, u] = [0, 1]^n = [\mathbf{0}, \mathbf{1}]$ and remark that a nonsingular, affine transformation (similar to the one for the moment curve) can be used to generalize the results.

We follow the work of Burer and Letchford [BL09] to present their characterization of $Q_{[0,1]}[f]$ which is mainly based on semidefinite programming (SDP) and the *Boolean quadric polytope* (cf. [BL09] and references therein). The key observation for an SDP-relaxation of $Q_{[0,1]}[f]$ is that

$$\begin{pmatrix} 1 \\ x \end{pmatrix}\begin{pmatrix} 1 \\ x \end{pmatrix}^\intercal = \begin{pmatrix} 1 & x^\intercal \\ x & xx^\intercal \end{pmatrix} \in \text{PSD} \quad \text{for all } x \in \mathbf{R}^n,$$

where $(xx^\intercal)_{i,j} = x_i x_j$ and $X \in \text{PSD}$ denotes that X belongs to the *convex* set of positive semidefinite matrices. To embed $Q_{[0,1]}[f]$ in this context, a symmetric matrix $Z \in \mathbf{R}^{n \times n}$ is introduced, where $Z_{i,j} = z_{i,j}$, such that all $(x, z) \in Q_{[0,1]}[f]$ necessarily satisfy that

$$\hat{Z} := \begin{pmatrix} 1 & x^\intercal \\ x & Z \end{pmatrix} \in \text{PSD}.$$

A second class of constraints of $Q_{[0,1]}[f]$ are the facets of the Boolean quadric polytope which is given as

$$\mathcal{B} := \operatorname{conv}\left\{(x, z) \in \{0,1\}^{n+\binom{n}{2}} \mid z_{i,j} = x_i x_j \text{ for all } 1 \le i < j \le n, x \in \{0,1\}^n\right\}.$$

Let $z_{\mathcal{B}}$ denote the subvector of z consisting only of the components $z_{i,j}$ with $1 \le i < j \le n$. From the definition of $\mathcal{B}$ and $Q_{[0,1]}[f]$ it becomes obvious that $\mathcal{B}$ is the projection of $Q_{[0,1]}[f]$ onto the $(x, z_{\mathcal{B}})$-space and thus, all valid constraints for $\mathcal{B}$ are also valid for $Q_{[0,1]}[f]$. In particular, large classes of facets of $\mathcal{B}$ are also facets of $Q_{[0,1]}[f]$. However, the facet-description of $\mathcal{B}$ is not known, in general, but in dimension $n = 2$ the Boolean quadric polytope $\mathcal{B}$ is described by the first level RLT constraints (see Section 4.1) which are equivalent to the convex hull of $x_1 x_2$ (see Example 3.14), i.e.,

$$z_{i,j} \le x_i, \quad z_{i,j} \le x_j, \quad z_{i,j} \ge 0, \quad z_{i,j} \ge x_i + x_j - 1. \tag{1RLT}$$

We say $\hat{Z} \in$ 1RLT if and only if the submatrix Z satisfies all constraints in Equation (1RLT). With the help of these relaxations, $Q_{[0,1]}[f]$ can be completely characterized for $n = 2$.

Theorem 5.6 (Theorem 2 in [AB10]). *Let $n = 2$. Then,*

$$Q_{[0,1]}[(x_1^2, x_2^2, x_1x_2)] = \left\{(x,z) \in \mathbf{R}^{2+3} \mid \hat{Z} \in \text{PSD} \cap \text{1RLT}\right\}.$$

There is also a complete description of $Q_{[0,1]}[f]$ for $n = 3$ which is based on so-called *doubly nonnegative matrices*. See [AB10] for details. For larger dimensions n an explicit description for $Q_{[0,1]}[f]$ is not known. Yet, computational experiments in [Ans09] for dimension $n = 30$ show that the relaxation quality with the constraint $\hat{Z} \in \text{PSD} \cap \text{1RLT}$ is still reasonable. The gap of the considered programs with quadratic objective functions and box constraints is always less than 4%.

Burer and Letchford [BL09] provide a very general, but computationally not tractable characterization for $Q_{[0,1]}[f]$, which is a special case of our subsequent results: They classify all valid inequalities $a^\intercal x + \gamma \leq \alpha^\intercal z = \sum_{1 \leq i \leq j \leq n} \alpha_{i,j} z_{i,j}$ for $Q_{[0,1]}[f]$ in terms of the convexity behavior of the corresponding function $\alpha^\intercal f$.

Proposition 5.7 (Proposition 8 in [BL09]). *Suppose that $a^\intercal x + \gamma \leq \alpha^\intercal z$ is valid for $Q_{[0,1]}[f]$ and $\alpha^\intercal f$ is **convex**. Then, $a^\intercal x + \gamma \leq \alpha^\intercal z$ is valid for the following convex set*

$$\left\{(x,z) \in [0,1]^{n+\binom{n+1}{2}} \mid \hat{Z} \in \mathit{PSD}\right\}.$$

Proposition 5.8 (Proposition 9 in [BL09]). *Suppose that $a^\intercal x + \gamma \leq \alpha^\intercal z$ is valid for $Q_{[0,1]}[f]$ and $\alpha^\intercal f$ is **concave**. Then, $a^\intercal x + \gamma \leq \alpha^\intercal z$ is valid for the following polytope*

$$\left\{(x,z) \in [0,1]^{n+\binom{n+1}{2}} \mid (x, z_{\mathcal{B}}) \in \mathcal{B},\ z_{i,i} \leq x_i \text{ for all } 1 \leq i \leq n\right\}.$$

Proposition 5.9 (Proposition 10 and Corollary 3 in [BL09]). *For $n \geq 2$, let $\mathcal{V}$ be the collection of all (a, α, γ) such that $a^\intercal x + \gamma \leq \alpha^\intercal z$ is valid for $Q_{[0,1]}[f]$ and $\alpha^\intercal f$ **indefinite**. Then,*

$$Q_{[0,1]}[f] = \left\{ (x,z) \in [0,1]^{n+\binom{n+1}{2}} \;\middle|\; \begin{array}{l} \hat{Z} \in \text{PSD}, \\ (x, z_{\mathcal{B}}) \in \mathcal{B},\ z_{i,i} \leq x_i \text{ for all } 1 \leq i \leq n, \\ a^\intercal x + \gamma \leq \alpha^\intercal z \text{ for all } (a, \alpha, \gamma) \in \mathcal{V} \end{array} \right\}.$$

Propositions 5.7 through 5.9 indicate that all constraints needed to describe $Q_{[0,1]}[f]$ can be categorized by the convexity pattern of the function $\alpha^\intercal f$. In the following sections we show that these results hold for general vectors of functions f and that the set of valid constraints for $Q_{[0,1]}[f]$ can be obtained from the convex envelopes of $\alpha^\intercal f$, $\alpha \in \mathbf{R}^m$.

5.1.3. General Functions: Inclusion Certificates

Recently, Tawarmalani [Taw10] presented first results for the simultaneous convex hull of general functions over continuous domains. The author considers even a more general setting, namely the convex hull of the set

$$M^D := \{(x, z) \mid H(x) \leq z \leq F(x), x \in D\},$$

where $H = (h_1(x), \ldots, h_m(x))$ with $h_i : \mathbf{R}^n \to \mathbf{R} \cup \{-\infty\}$, $i = 1, \ldots, m$, and $F = (f_1(x), \ldots, f_m(x))$ with $f_i : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$, $i = 1, \ldots, m$, and $D \subseteq \mathbf{R}^n$ is a compact set. It is assumed that either $h_i(x) = -\infty$ ($f_i(x) = +\infty$) for all $x \in D$ or h_i (f_i) exhibits an affine minorant (majorant), and either $h_i > -\infty$ or $f_i < +\infty$ for all $i = 1, \ldots, m$, so that there is no line in M^D.

The main focus of Tawarmalani's work is to reduce the set of possible extreme points of $\operatorname{conv}(M^D)$. If this set can be expressed as disjunctive union of subsets, then $\operatorname{conv}(M^D)$ may be relaxed easier when restricted to these subsets and disjunctive programming techniques can be used to describe $\operatorname{conv}(M^D)$ in an extended space.

Example 5.10. Given the set $M^D = \{(x, y, z) \in \mathbf{R}^{1+1+2} \mid z_1 \geq h_1(x, y), z_2 = h_2(x, y), x \in [l_x, u_x], y \in [l_y, u_y]\}$. If h_2 is component-wise linear in x and h_1 is component-wise concave in x, $\operatorname{conv}(M^D)$ can be represented as

$$\operatorname{conv}(M^D) = \operatorname{conv}\left(\bigcup_{x \in \{l_x, u_x\}} \{(x, y, z) \mid z_1 \geq h_1(x, y), z_2 = h_2(x, y), y \in [l_y, u_y]\} \right). \quad (5.3)$$

This is a well-known approach that we also applied in Chapter 4. The representation in Equation (5.3) allows to compute the convex hulls over the subsets $D_l := \{l_x\} \times [l_y, u_y]$ and $D_u := \{u_x\} \times [l_y, u_y]$ separately. For simplicity assume that $\operatorname{conv}(M^{D_l}) = \{v = (x, y, z)^\intercal \mid A_l v \leq b_l\}$ and $\operatorname{conv}(M^{D_u}) = \{v = (x, y, z)^\intercal \mid A_u v \leq b_u\}$. Then, an extended space descrip-

tion for $\text{conv}(M^D)$ is given by

$$\begin{aligned}&\left\{v \mid v = \lambda_l v^l + \lambda_u v^u, A_l v^l \leq b_l, A_u v^u \leq b_u, \lambda_l + \lambda_u = 1, \lambda_l, \lambda_u \geq 0\right\}\\ =&\left\{v \mid v = \tilde{v}^l + \tilde{v}^u, A_l \tilde{v}^l \leq \lambda_l b_l, A_u \tilde{v}^u \leq \lambda_u b_u, \lambda_l + \lambda_u = 1, \lambda_l, \lambda_u \geq 0\right\}.\end{aligned}$$

◇

To identify extreme points or to exclude potential candidates, Tawarmalani adapts the following well-known criteria: A point (x^0, z^0) is no extreme point of $\text{conv}(M^D)$ if it can be represented as convex combination of points different from (x^0, z^0), i.e., $(x^0, z^0) = \sum_k \lambda_k (x^k, z^k)$, $(x^0, z^0) \neq (x^k, z^k) \in M^D$, $1 = \sum_k \lambda_k$, and $\lambda_k \geq 0$ for all k. Note that we can restrict the points (x^k, z^k) to be of the form $(x, H(x))$ or $(x, F(x))$ with $x \in D$ since these points are the potential extreme points of $\text{conv}(M^D)$. Based on this, Tawarmalani suggests alternatively to express the convex combination by a probability measure $\mu(x^0)$ with support x^k and probabilities λ_k such that its expectation yields $E_{\mu(x^0)}[x] = \sum_k \lambda_k x^k = x^0$. Then, $E_{\mu(x^0)}[H(x)] = \sum_k \lambda_k H(x^k)$, for instance.

Definition 5.11. Let D be a compact set. For each point $x^0 \in D'$ with $D' \subseteq \text{conv}(D)$, an *inclusion certificate* is a measure $\mu(x^0)$ with its support in D such that $E_{\mu(x^0)}[x] = x^0$.

In this setting a point x^0 is a proper convex combination of points $x^k \neq x^0$ if and only if $\mu(x^0)$ is not a *Dirac measure*, i.e., $\mu(x^k) = 0$ for all $k \neq 0$ and $\mu(x^0) = 1$. The next result states sufficient conditions for a subset $X \subseteq D$ such that it is not contained in the projection of the extreme points of $\text{conv}(M^D)$ onto the x-space.

Theorem 5.12 (Theorem 2.1 in [Taw10])**.** *Let $X \subseteq D$ be such that for each $x^0 \in X$ there exists a non Dirac measure $\mu(x^0)$, that satisfies the following conditions*

1. $E_{\mu(x^0)}[x] = x^0$,
2. $E_{\mu(x^0)}[H(x)] \leq H(x^0) \leq F(x^0) \leq E_{\mu(x^0)}[F(x)]$.

Then, $\text{cl conv}(M^D) = \text{conv}(M^D) = \text{conv}(M^{D \setminus X})$. *Further,* $\text{cl conv}(M^D)$ *does not contain any lines and the projection of the extreme points of* $\text{conv}(M^D)$ *onto the space of x variables does not intersect with X.*

This result follows immediately if both $h_i > -\infty$ and $f_i < \infty$, so that $\text{conv}(M^D)$ is not only closed but also bounded. Then, (x, z) is no extreme

point of $\text{conv}(M^D)$ if there are points $x^k \in D$ different from x and convex multipliers $\lambda_k \geq 0$ such that $x = \sum_k \lambda_k x^k$, $1 = \sum_k \lambda_k$, and $\sum_k \lambda_k H(x^k) \leq z \leq \sum_k \lambda_k F(x^k)$. In other words, the inclusion certificate proves exclusion of a point from the set of extreme points.

Tawarmalani applies inclusion certificates to show that the convex hull of the graph of multilinear terms over a box $[l, u] \subseteq \mathbf{R}^n$ is generated by the vertices of this box. Moreover, compositions of variable disjoint functions are identified for which the inclusion certificates of the individual functions are also valid for the composition. See Theorem 3.6 in [Taw10]. This result leads to the following useful corollary.

Corollary 5.13 (Corollary 3.9 in [Taw10])**.** *Consider $H(x) : D \to \mathbf{R}^m \cup \{-\infty\}$ and $F(x) : D \to \mathbf{R}^m \cup \{\infty\}$ restricted to the compact domain $D \subseteq \mathbf{R}^n$. For each $x^0 \in \text{conv}(D)$, let $\mu(x^0)$ be the inclusion certificate associated with the convex envelopes of h_i and concave envelopes of f_i, i.e., $\mu(x^0)$ has its support in D and $E_{\mu(x^0)}(h_i(x)) = \text{vex}_{\text{conv}(D)}[h_i](x^0)$ and $E_{\mu(x^0)}(f_i(x)) = \text{cave}_{\text{conv}(D)}[f_i](x^0)$ for all $i \in \{1, \ldots, m\}$. Let*

$$C^D = \{(x, z) \mid \text{vex}_{\text{conv}(D)}[h_i](x) \leq z \leq \text{cave}_{\text{conv}(D)}[f_i](x), i = 1, \ldots, m, x \in D\}$$

and $M^D = \{(x, z) \mid H(x) \leq z \leq F(x), x \in D\}$. Then, $C^{\text{conv}(D)} = \text{conv}(M^D)$. In particular, the relation holds, if D is a polytope, f_i and h_i have polyhedral envelopes for all i, and the polyhedral subdivisions of D associated with $\text{vex}_D[h_i]$ and $\text{cave}_D[f_i]$ are the same.

This corollary can be applied for functions $h_i : [l, u] \subseteq \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$, which are submodular restricted to the vertices of $D = [l, u]$ and whose convex envelopes are polyhedral. Note that submodular functions share the same inclusion certificate because the polyhedral subdivisions of D associated with $\text{vex}_D[h_i]$ are identical for all i and correspond to Kuhn's triangulation (see Section 3.1.1 and [TRX12]). This implies that the convex hull $\mathcal{E}_D[h]$ of the epigraph of submodular functions $h = (h_1, \ldots, h_m)$ is obtained by intersecting the convex hulls of the individual epigraphs.

A dual version of the concept of inclusion certificates, and especially Theorem 5.12, is deduced via the following program

$$L^D(a, \alpha): \quad \inf\{a^\intercal x + (\alpha^+)^\intercal H(x) + (\alpha^-)^\intercal F(x) \mid x \in D\},$$

where $(a, \alpha) \in \mathbf{R}^{n+m}$, $\alpha_i^+ = \max\{\alpha_i, 0\}$ and $\alpha_i^- = \min\{\alpha_i, 0\}$. The nonconvex problem $L^D(a, \alpha)$ returns the same optimal objective function value as the

convex program

$$H^D(a,\alpha): \quad \inf\{a^\intercal x + (\alpha^+)^\intercal z + (\alpha^-)^\intercal z \mid (x,z) \in \operatorname{conv}(M^D)\}$$

since the extreme points of $\operatorname{conv}(M^D)$ are of the form $(x, H(x))$ or $(x, F(x))$. Therefore, an optimal solution x^* of $L^D(a,\alpha)$ corresponds to a potential extreme point $(x^*, H(x^*))$ or $(x^*, F(x^*))$ of $\operatorname{conv}(M^D)$. Moreover, let γ denote the optimal objective function value of $L^D(a,\alpha)$, then the constraint $\gamma \leq a^\intercal x + (\alpha^+)^\intercal z + (\alpha^-)^\intercal z$ corresponds to a supporting hyperplane on $\operatorname{conv}(M^D)$. If x^* is the unique solution of $L^D(a,\alpha)$, either $(x^*, H(x^*))$ or $(x^*, F(x^*))$ satisfies the definition of an *exposed point* of $\operatorname{conv}(M^D)$. An exposed point x is an extreme point of a convex set for which a supporting hyperplane exists such that the intersection of the convex set and the hyperplane reduces to $\{x\}$ (cf. [HUL01]).

Corollary 5.14 (Corollary 2.4 in [Taw10])**.** *Assume that $H(x) \leq F(x)$ for all $x \in D$. Let $X \subseteq D$ and assume that for all $x' \in X$ there is no $(a,\alpha) \in \mathbf{R}^{n+m}$ such that x' is the unique minimizer of $L^D(a,\alpha)$. Then,* $\operatorname{conv}(M^D) = \operatorname{cl}\operatorname{conv}(M^{D\setminus X})$. *More generally, if for all (a,α) that have a unique minimizer in $L^D(a,\alpha)$, it is true that $L^D(a,\alpha) = L^{D\setminus X}(a,\alpha)$, then* $\operatorname{conv}(M^D) = \operatorname{cl}\operatorname{conv}(M^{D\setminus X})$.

The uniqueness of the minimizer in the corollary can be neglected if $L^D(a,\alpha) = L^{D\setminus X}(a,\alpha)$ for all $(a,\alpha) \in \mathbf{R}^{n+m}$. Then, the sets M^D and $M^{D\setminus X}$ are characterized by identical supporting hyperplanes so that $\operatorname{conv}(M^D) = \operatorname{cl}\operatorname{conv}(M^D) = \operatorname{cl}\operatorname{conv}(M^{D\setminus X})$ (see Remark 2.5, Corollary 2.6, and the comments afterwards in [Taw10]). We illustrate the results by a simple example.

Example 5.15. Consider the continuously differentiable function $f : \mathbf{R} \to \mathbf{R}$ over $D := [-1,1]$ which is defined by $f(x) := 0$ for all $x \leq 0$ and $f(x) := x^2$ for all $x > 0$. Let $M^D := \{(x,z) \in \mathbf{R}^{1+1} \mid f(x) \leq z \leq f(x)\}$ with $\operatorname{conv}(M^D) = \{(x,z) \in \mathbf{R}^{1+1} \mid f(x) \leq z \leq 1/2\, x + 1/2\} = Q_{[-1,1]}[f]$ (see Figure 5.2). The set of extreme points is given by $\{(x, f(x)) \mid x \in \{-1\} \cup [0,1]\}$ and the only extreme point which is not an exposed point is $(0, f(0))$. Let $X := \{-1\} \cup (0,1]$. It follows that $\operatorname{conv}(M^{D\setminus X}) \subsetneq \operatorname{conv}(M^D) = \operatorname{cl}\operatorname{conv}(M^{D\setminus X})$ and $L^D(a,\alpha) = L^{D\setminus X}(a,\alpha)$ for all $(a,\alpha) \in \mathbf{R}^{n+m}$.

$\diamond$

In the context of the simultaneous convex hull $Q_D[f]$ we have $H(x) = F(x) = f(x)$ so that the program $L^D(a,\alpha)$ evolves to

$$\tilde{L}^D(a,\alpha): \quad \inf\{a^\intercal x + \alpha^\intercal f(x) \mid x \in D\}.$$

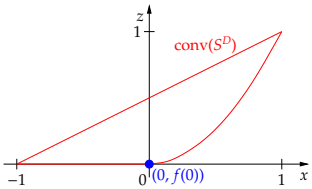


Figure 5.2.: The point $(0, f(0))$ is an extreme point but no exposed point of $\text{conv}(M^D)$.

The determination of the exposed and/or extreme points of $\text{conv}(M^D) = Q_D[f]$ via the program $\tilde{L}^D(a, \alpha)$ is a natural approach. It requires, however, to solve infinitely many auxiliary, nonconvex problems. To simplify these problems, the key idea in our approach is to exploit the knowledge of convex envelopes. For this, assume that γ is the optimal solution of $\tilde{L}^D(a, \alpha)$ for given $(a, \alpha) \in \mathbf{R}^{n+m}$ so that $\gamma \leq a^\intercal x + \alpha^\intercal z$ or equivalently $-a^\intercal x + \gamma \leq \alpha^\intercal z$ is a valid inequality for $Q_D[f]$. We show in the following section that for all valid inequalities it holds that $-a^\intercal x + \gamma \leq \text{vex}_D[\alpha^\intercal f](x) \leq \alpha^\intercal z$ and use this relation to derive the extreme points of $Q_D[f]$.

5.2. Basic Properties

In this section we link the simultaneous convex hull to the theory of convex envelopes and deduce basic properties of $Q_D[f]$. Initially, the extreme points of $Q_D[f]$ are described and then valid constraints are characterized. The corresponding results provide an inner as well as an outer description for $Q_D[f]$.

5.2.1. The Generating Set

For closed sets $S \subseteq \mathbf{R}^n$ it holds that the extreme points of $\text{conv}(S)$ are a subset of S and thus, the extreme points of the simultaneous convex hull $Q_D[f] = \text{conv}\left(\{(x, f(x)) \in \mathbf{R}^{n+m} \mid x \in D\}\right)$ are a subset of the set $\{(x, f(x)) \in \mathbf{R}^{n+m} \mid x \in D\}$. This observation motivates the analysis of $Q_D[f]$ in terms of the x variables for which we provide some definitions analogous to the concepts for convex envelopes (see Definitions 3.4 and 3.8).

Definition 5.16. The *generating set* of $Q_D[f]$ is defined as

$$\mathcal{G}(Q_D[f]) := \{x \in \mathbf{R}^n \mid (x, f(x)) \text{ is an extreme point of } Q_D[f]\}.$$

Assume that D is a polytope. The simultaneous convex hull $Q_D[f]$ is called *polyhedral* if its generating set is finite. It is called *vertex polyhedral* if $\mathcal{G}(Q_D[f]) = \text{vert}(D)$. The terms are analogously defined for $\mathcal{E}_D[f]$.

Corollary 5.14 by Tawarmalani implies that for every $x \in \mathcal{G}(Q_D[f])$ corresponding to an exposed point $(x, f(x))$ of $Q_D[f]$ there exists $(a, \alpha) \in \mathbf{R}^{n+m}$ such that x is the unique minimizer of the function $a^\intercal x + \alpha^\intercal f(x)$ over D or equivalently that $(x, f(x))$ is the unique minimizer of $a^\intercal x + \alpha^\intercal z$ over $Q_D[f]$. This also implies that x is a generator of the convex envelope of $\alpha^\intercal f$, i.e., $x \in G_D^{\text{vex}}[\alpha^\intercal f]$. Conversely, if $x \in G_D^{\text{vex}}[\alpha^\intercal f]$, then $x \in \mathcal{G}(Q_D[f])$.

Lemma 5.17. *Let $f : D \subseteq \mathbf{R}^n \to \mathbf{R}^m$ be continuous over the compact, convex set D. Then,*

$$\text{(i)} \bigcup_{\alpha \in \mathbf{R}^m} G_D^{\text{vex}}[\alpha^\intercal f] \subseteq \mathcal{G}(Q_D[f]) \quad \textit{and} \quad \text{(ii)} \ \mathcal{G}(Q_D[f]) \subseteq \text{cl}\left(\bigcup_{\alpha \in \mathbf{R}^m} G_D^{\text{vex}}[\alpha^\intercal f] \right).$$

Proof. To prove (i), consider a fixed $\alpha \in \mathbf{R}^m$ and choose $\bar{x} \in G_D^{\text{vex}}[\alpha^\intercal f]$. Assume that $\bar{x} \notin \mathcal{G}(Q_D[f])$ which implies the existence of $x^i \in D$, $x^i \neq \bar{x}$, and $\lambda_i \geq 0$ with $\sum_i \lambda_i = 1$ such that $(\bar{x}, f(\bar{x})) = \sum_i \lambda_i (x^i, f(x^i))$. Then, $\sum_i \lambda_i \alpha^\intercal f(x^i) = \alpha^\intercal (\sum_i \lambda_i f(x^i)) = \alpha^\intercal f(\bar{x})$. Thus, it follows that $(\bar{x}, \alpha^\intercal f(\bar{x})) = \sum_i \lambda_i (x^i, \alpha^\intercal f(x^i))$ which contradicts that $\bar{x} \in G_D^{\text{vex}}[\alpha^\intercal f]$.

Note that the ideas to prove (ii) are similar to the proof of Corollary 5.14. But as the corresponding paper [Taw10] is not reviewed yet, we give our own proof here. We assume that $\bar{x} \in \mathcal{G}(Q_D[f])$ and $(\bar{x}, f(\bar{x}))$ is an exposed point of $Q_D[f]$, i.e., there are $a \in \mathbf{R}^n$ and $\alpha \in \mathbf{R}^m$ so that $(\bar{x}, f(\bar{x}))$ is the unique minimizer of the linear function $a^\intercal x + \alpha^\intercal z$ over $Q_D[f]$. Assume that $\bar{x} \notin G_D^{\text{vex}}[\alpha^\intercal f]$. Then, there are $x^i \in D$, $x^i \neq \bar{x}$, and $\lambda_i \geq 0$ with $\sum_i \lambda_i = 1$ such that $(\bar{x}, \alpha^\intercal f(\bar{x})) = \sum_i \lambda_i (x^i, \alpha^\intercal f(x^i))$. We show that the latter statement is not true. From $(a^\intercal x^i + \alpha^\intercal f(x^i)) > (a^\intercal \bar{x} + \alpha^\intercal f(\bar{x}))$ for all x^i, we obtain $\sum_i \lambda_i (a^\intercal x^i + \alpha^\intercal f(x^i)) > (a^\intercal \bar{x} + \alpha^\intercal f(\bar{x}))$. With $\bar{x} = \sum_i \lambda_i x^i$ the later inequality evolves to $\sum_i \lambda_i (\alpha^\intercal f(x^i)) > \alpha^\intercal f(\bar{x})$ which contradicts $\alpha^\intercal f(\bar{x}) = \sum_i \lambda_i \alpha^\intercal f(x^i)$.

It remains to discuss $\bar{x} \in \mathcal{G}(Q_D[f])$, where $(\bar{x}, f(\bar{x}))$ is an extreme point but no exposed point of $Q_D[f]$. In this case there is no supporting hyperplane on $Q_D[f]$ such that $(\bar{x}, f(\bar{x}))$ is a unique minimizer. Therefore, $\bar{x}$ is not

necessarily in $\bigcup_{\alpha\in\mathbf{R}^m} G_D^{\text{vex}}[\alpha^\intercal f]$. Strasziewicz's Theorem (cf. [Roc70]) states that exposed points of a closed convex set in $\mathbf{R}^n$ are dense in the extreme points of this set. This property remains true for projections or subsets. Let S_{exps} denote the subset of $\mathcal{G}(Q_D[f])$, where for all $x \in S_{exps}$ the point $(x, f(x))$ is an exposed point of $Q_D[f]$. Then, S_{exps} is dense in $\mathcal{G}(Q_D[f])$. We showed that $S_{exps} \subseteq \bigcup_{\alpha\in\mathbf{R}^m} G_D^{\text{vex}}[\alpha^\intercal f]$ which implies $\mathcal{G}(Q_D[f]) = \text{cl}(S_{exps}) \subseteq \text{cl}(\bigcup_{\alpha\in\mathbf{R}^m} G_D^{\text{vex}}[\alpha^\intercal f])$ and concludes the proof. □

Remark 5.18. It is not clear if there really is a gap between $\bigcup_{\alpha\in\mathbf{R}^m} G_D^{\text{vex}}[\alpha^\intercal f]$ and $\mathcal{G}(Q_D[f])$. So far we are not aware of a counterexample.

In the following we illustrate possible applications of Lemma 5.17 in order to determine the generating set of $Q_D[f]$.

Example 5.19. Let $f : D \to \mathbf{R}^2$, $D := [l, u] \subseteq \mathbf{R}^2$, with $f_1(x) := x_1^2 + x_1x_2$ and $f_2(x) := x_1x_2 + x_2^2$. We show that the generating set $\mathcal{G}(Q_D[f])$ equals the boundary of D. Consider an arbitrary $\alpha \in \mathbf{R}^2$ and the corresponding function $g^\alpha := \alpha^\intercal f$ whose Hessian equals

$$H_{g^\alpha}(x) = \begin{pmatrix} 2\alpha_1 & \alpha_1 + \alpha_2 \\ \alpha_1 + \alpha_2 & 2\alpha_2 \end{pmatrix}.$$

The determinant of the Hessian is given by $-(\alpha_1 - \alpha_2)^2$. Therefore, the Hessian cannot be positive semidefinite so that there is a concave direction at each interior point of D which can be used as an underestimating segment. We can thus infer from Observation 3.5 that the generating set $G_D^{\text{vex}}[g^\alpha]$ is always a subset of the boundary of D denoted by $\text{bd}(D)$.

To show that $\bigcup_{\alpha\in\mathbf{R}^m} G_D^{\text{vex}}[g^\alpha] = \text{bd}(D)$, we consider two corner cases. For $\alpha = (1, 0)$ the function $g^\alpha = f_1$ is strictly convex along faces with $y \in \{l_y, u_y\}$. Using Observation 3.6, we conclude $G_D^{\text{vex}}[g^\alpha] = ([l_1, u_1] \times \{l_2\}) \cup ([l_1, u_1] \times \{u_2\})$. Analogously, for $\alpha = (0, 1)$ we obtain $G_D^{\text{vex}}[g^\alpha] = (\{l_1\} \times [l_2, u_2]) \cup (\{u_1\} \times [l_2, u_2])$. Then, $\bigcup_{\alpha\in\mathbf{R}^m} G_D^{\text{vex}}[g^\alpha] = \text{bd}(D) = \text{cl}(\text{bd}(D))$ which means $\mathcal{G}(Q_D[f]) = \text{bd}(D)$.

The geometric reason for $\mathcal{G}(Q_D[f]) = \text{bd}(D)$ is the linearity of both f_1 and f_2 over the line $x_2 = -x_1$ which implies that for all interior points x of D there are points x^1, x^2 in the boundary of D and $\lambda \in (0, 1)$ such that $(x, f_1(x), f_2(x)) = \lambda(x^1, f_1(x^1), f_2(x^1)) + (1 - \lambda)(x^2, f_1(x^2), f_2(x^2))$. ◇

A second application of Lemma 5.17 is the next result.

Corollary 5.20. *Let $f : D \subseteq \mathbf{R}^n \to \mathbf{R}^m$, where f is continuous and D is a compact, convex set. If there is an $\alpha \in \mathbf{R}^m$ such that $\alpha^\intercal f$ is strictly convex over D, then $\mathcal{G}(Q_D[f]) = D$.*

Proof. As $\alpha^\intercal f$ is strictly convex, $G_D^{\text{vex}}[\alpha^\intercal f] = D$. With Lemma 5.17 we can conclude $D = G_D^{\text{vex}}[\alpha^\intercal f] \subseteq \mathcal{G}(Q_D[f]) \subseteq D$ and thus, $\mathcal{G}(Q_D[f]) = D$. □

The converse of Corollary 5.20 is true for a vector of quadratic functions. This is already indicated in Example 5.19, where $\mathcal{G}(Q_D[f]) \neq D$ and there is no $\alpha \in \mathbf{R}^m$ such that $\alpha^\intercal f$ is strictly convex. However, for general functions the converse is not true as illustrated in the following example.

Example 5.21. Let $f : D \to \mathbf{R}^2$, $D := [0,1]$, with $f_1(x) := -2.0942\,x^3 + 5.807\,x^4 - 6.2334\,x^5 + 3.0486\,x^6 - 0.569\,x^7 + 0.25$ and $f_2(x) := 1.048\,x^3 - 0.72\,x^4 - 0.164\,x^5 + 0.161\,x^6$ which are depicted together with their convex envelopes in Figure 5.3. The convex envelopes read

$$\text{vex}_D[f_1](x) = \begin{cases} -0.10\,x + 0.250, & x < 0.508, \\ f_1(x), & x \geq 0.508, \end{cases}$$

$$\text{vex}_D[f_2](x) = \begin{cases} f_2(x), & x \leq 0.628, \\ 0.493\,x - 0.168, & x > 0.628. \end{cases}$$

The generating sets of the convex envelopes are $G_D^{\text{vex}}[f_1] = \{0\} \cup [0.508, 1]$ and $G_D^{\text{vex}}[f_2] = [0, 0.628] \cup \{1\}$ whose union is $[0,1] = D$. Lemma 5.17 implies that $\mathcal{G}(Q_D[f]) = D$.

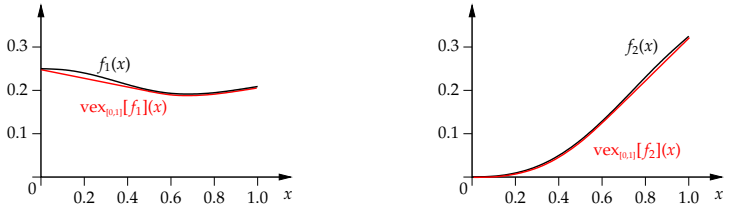


Figure 5.3.: The functions f_1 and f_2 (black) together with their convex envelopes (red).

But there is no $\alpha \in \mathbf{R}^2$ such that $\alpha^\intercal f$ is strictly convex over D: The eigenvalues of $g^\alpha := \alpha^\intercal f$ at $x \in \{0.1, 0.6, 0.9\}$ read $\frac{d^2 g^\alpha}{dx^2}(0.1) = -0.675\alpha_1 + 0.539\alpha_2$, $\frac{d^2 g^\alpha}{dx^2}(0.6) = 0.613\alpha_1 + 0.579\alpha_2$, and $\frac{d^2 g^\alpha}{dx^2}(0.9) = 0.146\alpha_1 - 0.561\alpha_2$. As neither f_1 nor f_2 is strictly convex, it is sufficient to normalize α and

consider $\alpha_1 \in \{-1, 1\}$. If $\alpha_1 = -1$, then $\frac{d^2 g^\alpha}{dx^2}(0.9) \geq 0$ if and only if $\alpha_2 \leq -0.26$, and $\frac{d^2 g^\alpha}{dx^2}(0.6) \geq 0$ if and only if $\alpha_2 \geq 1.05$, so that there is no $\alpha_2 \in \mathbf{R}$ yielding a convex function g^α. If $\alpha_1 = 1$, then $\frac{d^2 g^\alpha}{dx^2}(0.9) \geq 0$ if and only if $\alpha_2 \leq 0.26$, and $\frac{d^2 g^\alpha}{dx^2}(0.1) \geq 0$ if and only if $\alpha_2 \geq 1.25$, so that there is again no $\alpha_2 \in \mathbf{R}$ yielding a convex function g^α. Thus, there are no $\alpha \in \mathbf{R}^2$ such that g^α is convex or strictly convex.

◇

To conclude our considerations regarding the generating sets, we generalize Theorem 3.9, which deals with the polyhedrality of convex envelopes, to the higher dimensional objects $\mathcal{Q}_D[f]$ and $\mathcal{E}_D[f]$. Assume that the domain D is a polytope and f is a vector of continuously differentiable functions. Then, the following result allows us to focus on the vertices of D when checking whether $\mathcal{Q}_D[f]$ is polyhedral.

Theorem 5.22 (Generalization of Theorem 3.9)**.** *Let $f : D \subseteq \mathbf{R}^n \to \mathbf{R}^m$ be a vector of continuously differentiable functions and D be a polytope. The set $\mathcal{Q}_D[f]$ is polyhedral if and only if $\mathcal{Q}_D[f]$ is vertex polyhedral. The same is true for $\mathcal{E}_D[f]$.*

Proof. If $\mathcal{Q}_D[f]$ is vertex polyhedral, it is also polyhedral. For the converse direction we use some ideas of the proof of Lemma 5.17. As all extreme points of $\mathcal{Q}_D[f]$ are exposed points, we have that $\mathcal{G}(\mathcal{Q}_D[f]) = \bigcup_{\alpha \in \mathbf{R}^m} G_D^{\text{vex}}[\alpha^\top f]$. Moreover, polyhedrality of $\mathcal{Q}_D[f]$ induces finiteness of $\mathcal{G}(\mathcal{Q}_D[f])$ and thus, $G_D^{\text{vex}}[\alpha^\top f]$ is finite for all $\alpha \in \mathbf{R}^m$. Since $G_D^{\text{vex}}[\alpha^\top f]$ is finite, $\text{vex}_D[\alpha^\top f]$ is polyhedral. By Theorem 3.9, $\text{vex}_D[\alpha^\top f]$ is also vertex polyhedral and $G_D^{\text{vex}}[\alpha^\top f] = \text{vert}(D)$ for all $\alpha \in \mathbf{R}^m$. This implies $\mathcal{G}(\mathcal{Q}_D[f]) = \bigcup_{\alpha \in \mathbf{R}^m} G_D^{\text{vex}}[\alpha^\top f] = \text{vert}(D)$ and shows that $\mathcal{Q}_D[f]$ is vertex polyhedral. □

Similar to the work of Tawarmalani [Taw10], the derived properties of the generators can be used in a disjunctive programming approach to determine $\mathcal{Q}_D[f]$ in an **extended** space (see Example 5.10). This approach is appropriate if $\mathcal{Q}_D[f]$ restricted to the disjunctive subsets of extreme points can be explicitly described. Moreover, if the generating set is finite, combinatorial software like `QHull` [BDH96], `PORTA` [CL07], or `polymake` [GJ00] can be utilized to compute a complete outer description of $\mathcal{Q}_D[f]$. If the generating set is infinite, comparable algorithms are not available. To overcome these obstacles, we suggest an alternative approach based on

the knowledge of convex envelopes which directly aims at the computation of strong constraints in the **original** space of $Q_D[f]$.

5.2.2. Valid Inequalities

In this section we show that $(x,z) \in Q_D[f]$ if and only if $\mathrm{vex}_D[\alpha^\intercal f](x) \le \alpha^\intercal z$ for all $\alpha \in \mathbf{R}^m$. Our point of departure is the following elementary lemma.

Lemma 5.23. *Let $C \subseteq \mathbf{R}^{n+m}$ be a closed, convex set in the (x,z)-space $\mathbf{R}^{n+m}$ such that for all $\mu > 0$ and $(x,z) \in C$ it holds that $(x,\mu z) \in C$. Choose an arbitrary subset $V \subseteq C$ with $\mathrm{conv}(V) = C$. For each $\alpha \in \mathbf{R}^m_{\ge 0}$ we define $C_\alpha := \{(x,z) \in \mathbf{R}^{n+m} \mid (x,\alpha^\intercal z) \in \mathrm{conv}(V_\alpha)\}$, where $V_\alpha := \{(v,\alpha^\intercal w) \in \mathbf{R}^{n+1} \mid (v,w) \in V\}$. Then,*

$$C \quad = \quad \bigcap_{\alpha \in \mathbf{R}^m_{\ge 0}} C_\alpha.$$

Proof. "$\subseteq$": Let $(\bar{x},\bar{z}) \in C$. Then, $(\bar{x},\bar{z}) = \sum_k \lambda_k (v^k, w^k)$ for some $\lambda_k \ge 0$ with $\sum_k \lambda_k = 1$ and some $(v^k,w^k) \in V$. Thus, for each $\alpha \in \mathbf{R}^m_{\ge 0}$ we have $(v^k, \alpha^\intercal w^k) \in V_\alpha$ and

$$\alpha^\intercal \bar{z} = \sum_{j=1}^m \alpha_j \bar{z}_j = \sum_{j=1}^m \alpha_j \Big(\sum_k \lambda_k w_j^k \Big) = \sum_k \lambda_k \sum_{j=1}^m \alpha_j w_j^k = \sum_k \lambda_k \, (\alpha^\intercal w^k).$$

Therefore, $(\bar{x}, \alpha^\intercal \bar{z}) \in \mathrm{conv}(V_\alpha)$ and hence, $(\bar{x},\bar{z}) \in C_\alpha$.

"$\supseteq$": Assume that there is an $(\bar{x},\bar{z}) \in \Big(\bigcap_{\alpha \in \mathbf{R}^m_{\ge 0}} C_\alpha \Big) \setminus C$. As C is a closed, convex set, there exists a hyperplane $\{(x,z) \mid a^\intercal x + \alpha^\intercal z = \gamma\}$ for some $a \in \mathbf{R}^n$, $\alpha \in \mathbf{R}^m$, and $\gamma \in \mathbf{R}$, which separates the point $(\bar{x},\bar{z})$ from C (cf. [HUL01, Theorem 4.1.1]), i.e.,

$$a^\intercal x + \alpha^\intercal z \ge \gamma, \quad \text{for all } (x,z) \in C, \quad \text{and} \quad a^\intercal \bar{x} + \alpha^\intercal \bar{z} < \gamma.$$

Note that $\alpha \ge 0$ since the z-components are not bounded from above in C. In particular, $a^\intercal x + \alpha^\intercal z \ge \gamma$ is valid for all $(x,z) \in V \subseteq C$. Therefore, identifying $\alpha^\intercal z$ with a new variable $\tilde{z}$, we obtain that $a^\intercal x + \tilde{z} \ge \gamma$ is a valid inequality for $\mathrm{conv}(V_\alpha)$, but it is not satisfied by $(\bar{x}, \alpha^\intercal \bar{z})$. This shows that $a^\intercal x + \tilde{z} \ge \gamma$ separates $(\bar{x}, \alpha^\intercal \bar{z})$ from $\mathrm{conv}(V_\alpha)$ and implies that $a^\intercal x + \alpha^\intercal z \ge \gamma$ separates $(\bar{x},\bar{z})$ from C_α. This contradicts the assumption that $(\bar{x},\bar{z}) \in \bigcap_{\alpha \in \mathbf{R}^m} C_\alpha$. □

This lemma allows us to derive an alternative representation of the convex hull $\mathcal{E}_D[f]$ of the epigraph of a vector of functions in terms of convex envelopes. Throughout this chapter we assume that $D \subseteq \mathbf{R}^n$ is a full-dimensional, compact, convex set and that each function f_i, $i = 1, \dots, m$, is continuous over D. Our assumptions imply that $\mathcal{E}_D[f]$ is a closed, convex set so that we can describe $\mathcal{E}_D[f]$ with Lemma 5.23 by setting $C = \mathcal{E}_D[f] = \text{conv}(\{(x,z) \in \mathbf{R}^{n+m} \mid z \geq f(x), x \in D\})$ and $V = \{(x,z) \in \mathbf{R}^{n+m} \mid z \geq f(x), x \in D\}$.

Corollary 5.24. *Let $f : D \subseteq \mathbf{R}^n \to \mathbf{R}^m$, where f is continuous and D is a compact, convex set. Then,*

$$\begin{aligned}\mathcal{E}_D[f] &\overset{(a)}{=} \bigcap_{\alpha \in \mathbf{R}^m_{\geq 0}} \{(x,z) \in \mathbf{R}^{n+m} \mid (x, \alpha^\intercal z) \in \mathcal{E}_D[(\alpha^\intercal f)]\} \\ &\overset{(b)}{=} \bigcap_{\alpha \in \mathbf{R}^m_{\geq 0}} \{(x,z) \in \mathbf{R}^{n+m} \mid \alpha^\intercal z \geq \text{vex}_D[\alpha^\intercal f](x),\ x \in D\}.\end{aligned}$$

Proof. Equation (a) is a direct consequence of Lemma 5.23, where we set $C = \mathcal{E}_D[f]$, $V = \{(x,z) \in \mathbf{R}^{n+m} \mid z \geq f(x), x \in D\}$, and $V_\alpha = \{(x,w) \in \mathbf{R}^{n+1} \mid w \geq \alpha^\intercal f(x), x \in D\}$ so that $\text{conv}(V_\alpha) = \mathcal{E}_D[(\alpha^\intercal f)]$. Equation (b) follows from the fact $\mathcal{E}_D[(\alpha^\intercal f)] = \{(x,\tilde{z}) \in \mathbf{R}^{n+1} \mid \text{vex}_D[\alpha^\intercal f](x) \leq \tilde{z},\ x \in D\}$. □

The representation of $\mathcal{E}_D[f]$ in Corollary 5.24 is closely related to the supporting hyperplanes of $\mathcal{E}_D[f]$. For a given $\alpha \in \mathbf{R}^m_{\geq 0}$ the constraint $\alpha^\intercal z \geq \text{vex}_D[\alpha^\intercal f](x)$ comprises all $a \in \mathbf{R}^n$ and $\gamma \in \mathbf{R}$ so that $a^\intercal x + \alpha^\intercal z \geq \gamma$ is a valid inequality for $\mathcal{Q}_D[f]$ because $\alpha^\intercal z \geq \text{vex}_D[\alpha^\intercal f](x) \geq -a^\intercal x + \gamma$ for all $(x,z) \in \mathcal{Q}_D[f]$. Supporting hyperplanes or strong valid inequalities for $\mathcal{E}_D[f]$ are generally not known. Thus, Corollary 5.24 offers one possibility to determine $\mathcal{E}_D[f]$ by exploiting the knowledge of convex envelopes.

The description of $\mathcal{E}_D[f]$ in Corollary 5.24 can be used to derive a similar description for $\mathcal{Q}_D[f]$. For this, we link the two objects to each other. Note that for $m = 1$ the convex hull $\mathcal{Q}_D[f]$ can be described by

$$\begin{aligned}\mathcal{Q}_D[f] &= \{(x,z) \in \mathbf{R}^{n+1} \mid \text{vex}_D[f](x) \leq z,\ \text{cave}_D[f](x) \geq z,\ x \in D\} \\ &= \{(x,z) \in \mathbf{R}^{n+1} \mid \text{vex}_D[f](x) \leq z,\ \text{vex}_D[-f](x) \leq -z,\ x \in D\} \\ &= \{(x,z) \in \mathbf{R}^{n+1} \mid (x,z) \in \mathcal{E}_D[f]\} \cap \{(x,z) \in \mathbf{R}^{n+1} \mid (x,-z) \in \mathcal{E}_D[-f]\}.\end{aligned}$$

We generalize this representation to higher dimensions m. For this, we introduce the diagonal matrix I_β with entries $(I_\beta)_{i,i} = \beta_i$ for all $i = 1, \dots, m$. Then, for a given $z \in \mathbf{R}^m$ the product $I_\beta z$ results in a vector with $(I_\beta z)_i = \beta_i z_i$, $i = 1, \dots, m$. Using this notation, we verified that

$$Q_D[f] = \bigcap_{\beta \in \{-1,1\}^m} \left\{(x,z) \in \mathbf{R}^{n+m} \mid (x, I_\beta z) \in \mathcal{E}_D[I_\beta f]\right\}. \tag{5.4}$$

Therefore, one can describe $Q_D[f]$ by determining all $\mathcal{E}_D[I_\beta f]$, $\beta \in \{-1,1\}^m$. For instance, Tawarmalani [Taw10] proved for the vector of submodular functions $f(x) = (f_1, \dots, f_m)$ that $\mathcal{E}_D[I_\beta f]$ with $\beta = (1, \dots, 1)^\intercal$ is obtained by intersecting the convex hulls of the individual epigraphs $\{(x,z) \in \mathbf{R}^{n+m} \mid z_i \geq f_i(x), x \in D\}$. Alternatively, we use the representation of $Q_D[f]$ in Equation (5.4) to extend Corollary 5.24 for $Q_D[f]$.

Corollary 5.25. *Let $f : D \subseteq \mathbf{R}^n \to \mathbf{R}^m$, where f is continuous and D is a compact, convex set. Then,*

$$\begin{aligned} Q_D[f] &\overset{(a)}{=} \bigcap_{\alpha \in \mathbf{R}^m} \{(x,z) \in \mathbf{R}^{n+m} \mid (x, \alpha^\intercal z) \in Q_D[(\alpha^\intercal f)]\} \\ &\overset{(b)}{=} \bigcap_{\alpha \in \mathbf{R}^m} \{(x,z) \in \mathbf{R}^{n+m} \mid \alpha^\intercal z \geq \mathrm{vex}_D[\alpha^\intercal f](x),\ x \in D\}. \end{aligned}$$

From Corollaries 5.24 and 5.25 we obtain that $\mathcal{E}_D[f]$ and $Q_D[f]$ can be represented via lower-dimensional objects using the convex envelopes $\mathrm{vex}_D[\alpha^\intercal f]$, $\alpha \in \mathbf{R}^m$. Nevertheless, the representation implies the knowledge of $\mathrm{vex}_D[\alpha^\intercal f]$ for **all** $\alpha \in \mathbf{R}^m$. We thus address two natural questions in the following: Which $\alpha \in \mathbf{R}^m$ are actually needed in the description of $\mathcal{E}_D[f]$ and $Q_D[f]$? Which $\alpha \in \mathbf{R}^m$ generate tight relaxations of $\mathcal{E}_D[f]$ and $Q_D[f]$?

To answer the first question, we assume that $D = [l, u]$ and collect all α which either lead to a convex function $\alpha^\intercal f$ or to functions, whose convex envelope is vertex polyhedral and generated by the same triangulation. For this, let $\mathcal{T}$ denote the set of all triangulations T of D. Then,

$$C_{\mathrm{vex}} := \{\alpha \in \mathbf{R}^m \mid \alpha^\intercal f \text{ is convex}\},$$

$$C_{\mathrm{poly},T} := \left\{ \alpha \in \mathbf{R}^m \;\middle|\; \begin{array}{l} \mathrm{vex}_D[\alpha^\intercal f](x) \text{ is vertex polyhedral and its} \\ \text{polyhedral subdivision of } [l,u] \text{ corresponds to T} \end{array} \right\}$$

for all $T \in \mathcal{T}$. The sets C_{vex} and $C_{\text{poly},T}$ are cones which can be empty, e.g., $C_{\text{vex}} = \emptyset$ in Example 5.21. The motivation for the definition of these sets is the following: Let $\alpha^1, \alpha^2, \alpha \in C_{\text{vex}}$, for example, such that there are $\mu_1, \mu_2 > 0$ with $\mu_1\alpha^1 + \mu_2\alpha^2 = \alpha$. Recall that for convex functions $\alpha^\intercal f$ it holds that $\text{vex}_D[\alpha^\intercal f] = \alpha^\intercal f$. Then,

$$\mu_1 \text{vex}_D[(\alpha^1)^\intercal f] + \mu_2 \text{vex}_D[(\alpha^2)^\intercal f] = \mu_1(\alpha^1)^\intercal f + \mu_2(\alpha^2)^\intercal f = \alpha^\intercal f = \text{vex}_D[\alpha^\intercal f].$$

This implies that the constraint $\text{vex}_D[\alpha^\intercal f](x) \leq \alpha^\intercal z$ is a conic combination of the constraints $\text{vex}_D[(\alpha^i)^\intercal f](x) \leq (\alpha^i)^\intercal z$, $i = 1, 2$, and therefore, it is not needed in the description of $Q_D[f]$ by Corollary 5.25. A similar argumentation holds for $C_{\text{poly},T}$ and implies that the interior points of the cones are not necessary for $Q_D[f]$ (if the cones are closed). We obtain the following lemma.

Lemma 5.26. *Let $f : D \subseteq \mathbf{R}^n \to \mathbf{R}^m$, where f is continuous, $D = [l, u]$, and $\mathcal{T}$ denotes the set of triangulations of D. Assume that the cones C_{vex} and $C_{\text{poly},T}$, $T \in \mathcal{T}$, are closed and define $M := \mathbf{R}^m \setminus (\text{int}(C_{\text{vex}}) \cup \bigcup_{T\in\mathcal{T}} \text{int}(C_{\text{poly},T}))$. Then,*

$$Q_D[f] = \bigcap_{\alpha\in M} \{(x, z) \in \mathbf{R}^{n+m} \mid \alpha^\intercal z \geq \text{vex}_D[\alpha^\intercal f](x),\ x \in D\}.$$

In the next example we compute C_{vex} and $C_{\text{poly},T}$ for $f = (x_1^2, x_2^2, x_1x_2)^\intercal$ whose simultaneous convex hull was derived by Anstreicher and Burer [AB10] and was discussed in Section 5.1.2.

Example 5.27. Let $f : \mathbf{R}^2 \to \mathbf{R}^3$ with $f = (x_1^2, x_2^2, x_1x_2)^\intercal$ be restricted to the box $D := [0, 1]^2 \subseteq \mathbf{R}^2$. To determine C_{vex}, we analyze the Hessian of each function $\alpha^\intercal f = \alpha_1x_1^2 + \alpha_2x_2^2 + \alpha_3x_1x_2$, $\alpha \in \mathbf{R}^3$, which reads

$$H_{\alpha^\intercal f}(x) = \begin{pmatrix} 2\alpha_1 & \alpha_3 \\ \alpha_3 & 2\alpha_2 \end{pmatrix}.$$

Thus, $C_{\text{vex}} = \{\alpha \in \mathbf{R}^3 \mid H_{\alpha^\intercal f}(x) \succeq 0\}$ corresponds to the closed cone of all positive semidefinite matrices, whose boundary consists of all positive semidefinite matrices having at least one eigenvalue equal to zero. The constraints $\text{vex}_D[\alpha^\intercal f](x) \leq \alpha^\intercal z$ with $\alpha \in C_{\text{vex}}$ comprise all valid linear constraints $a^\intercal x + \gamma \leq \alpha^\intercal z$ for $Q_D[f]$ such that $\alpha^\intercal f$ is convex. According to

Proposition 5.8 these constraints are also valid for

$$\left\{(x,z) \in [0,1]^{2+\binom{2+1}{2}} \mid \hat{Z} \in \text{PSD}\right\}.$$

To determine the cones $C_{\text{poly},T}$ over $D = [0,1]^2$, we consider the two triangulations of $[0,1]^2$ resulting in $C_{\text{poly},T_1} = \text{cone}(\{(-1,0,0),(0,-1,0),(0,0,1))$ and $C_{\text{poly},T_2} = \text{cone}(\{(-1,0,0),(0,-1,0),(0,0,-1))$. Note that the union of C_{poly,T_1} and C_{poly,T_2} is again a cone. The constraints $\text{vex}_D[\alpha^\intercal f] \leq \alpha^\intercal z$ corresponding to the extreme rays of the cones read

$$\begin{array}{llll} (-1,0,0) & : -x_1 \leq -z_{1,1}, & (0,0,1) & : \max\{0, x_1 + x_2 - 1\} \leq z_{1,2}, \\ (0,-1,0) & : -x_2 \leq -z_{2,2}, & (0,0,-1) & : \max\{-x_1, -x_2\} \leq -z_{1,2}, \end{array}$$

which is equivalent to $z_{i,i} \leq x_i$, $i = 1,2$, and $(x, z_{1,2}) = (x, z_{\mathcal{B}}) \in \mathcal{B}$, where $\mathcal{B}$ is the Boolean quadric polytope introduced in Section 5.1.2. This observation extends Proposition 5.8, which states that all inequalities $a^\intercal x + \gamma \leq \alpha^\intercal z$, which are valid for $Q_{[0,1]}[f]$ and where $\alpha^\intercal f$ is concave, are also **valid** for the set

$$\left\{(x,z) \in [0,1]^{2+\binom{2+1}{2}} \mid (x, z_{\mathcal{B}}) \in \mathcal{B},\ z_{i,i} \leq x_i \text{ for all } 1 \leq i \leq n\right\}.$$

The inequalities $a^\intercal x + \gamma \leq \alpha^\intercal z$, which are valid for $Q_{[0,1]}[f]$ and where $\text{vex}_D[\alpha^\intercal f]$ is vertex polyhedral, are not only valid but describe this set **completely**. This strengthening is achieved because concave functions $\alpha^\intercal f$ are a subset of functions $\alpha^\intercal f$ whose convex envelope is vertex polyhedral. ◇

We observed in the example that the union of the cones $C_{\text{poly},T}$ over all triangulations T is a cone again. However, this is not true in general as we show in the following example.

Example 5.28. Let $f : D \subseteq \mathbf{R}^2 \to \mathbf{R}^2$ with $f_1(x) := (x_1^3 - 2x_1)(x_2^2 - 0.5)$, $f_2(x) := -0.18x_1x_2$, and $D := [-2,1] \times [-0.75, 0.95]$. In Example 4.11 we showed that f_1 is vertex polyhedral over D. The function f_2 is vertex polyhedral since it is component-wise concave. Assume that the convex envelope of $f_1 + f_2$ is vertex polyhedral so that $\text{vex}_D[f_1 + f_2](x)$ would be equivalent to

$$\text{vex}_D[f_1 + f_2](x) = \max\left\{\tfrac{1}{400}(79x_1 - 176x_2 - 182),\ \tfrac{1}{2000}(463x_1 - 760x_2 - 888)\right\}.$$

However, at the point $\bar{x} = (-0.74, -0.25)$ we observe $(f_1 + f_2)(\bar{x}) \approx -0.50 < -0.49 \approx \text{vex}_{\text{vert}(D)}[f_1 + f_2](\bar{x})$ which contradicts vertex polyhedrality of $\text{vex}_D[f_1 + f_2]$. In particular, this example shows that the convex enve-

lope of the sum of two functions, having a vertex polyhedral convex envelope, is not necessarily vertex polyhedral. ◇

5.3. Vectors of Univariate Convex Functions

A first step towards the application of the proposed concepts is presented in this section, where we explicitly derive the cones C_{vex} and C_{poly} for vectors f of two and three univariate convex functions satisfying specific assumptions. Note that we omit the index T in $C_{\text{poly},T}$, $T \in \mathcal{T}$, as there is only one triangulation of univariate boxes. The cones C_{vex} and C_{poly} are then used to determine necessary and sufficient sets of $\alpha \in \mathbf{R}^m$ such that $Q_D[f]$ is completely described via the constraints $\text{vex}_D[\alpha^\intercal f](x) \leq \alpha^\intercal z$. Based on this, we suggest a small set of α such that the corresponding constraints $\text{vex}_D[\alpha^\intercal f](x) \leq \alpha^\intercal z$ yield a tight relaxation of $Q_D[f]$.

Our analysis relies on the ability to describe convex envelopes of univariate functions $\alpha^\intercal f$, whose convexity behavior strongly depends on α. Although univariate functions may seem to be the most simple case, they often occur in the reformulation process of more complicated functions as indicated in Example 1.1. Their convex envelopes are only known for specific classes of univariate functions (e.g., see [LP03]), whereas no explicit descriptions for general functions are available. However, there are constructive procedures based on the location of the local extreme points and the inflection points [McC76, MF95], which we use to derive the convex envelopes of functions with one or two inflection points. This allows us to analyze vectors of two univariate convex functions, where $\alpha^\intercal f$ exhibits at most one inflection point, and vectors of three univariate convex functions,where $\alpha^\intercal f$ exhibits at most two inflection points, in Subsections 5.3.1 and 5.3.2, respectively.

5.3.1. Two Univariate Convex Functions

Let $f : [l, u] \subseteq \mathbf{R} \to \mathbf{R}^2$ be a vector of two univariate functions such that $\alpha^\intercal f$ possesses at most one inflection point over $[l, u]$ for all $\alpha \in \mathbf{R}^2$. Thus, there are four possible types of functions $\alpha^\intercal f$: (i) convex, (ii) concave, (iii) *convex-concave*, i.e., first strictly convex, then strictly concave, and (iv) *concave-convex*, i.e., first strictly concave, then strictly convex. The convex envelopes of the first two types are generally known. The convex envelope of a convex function is the function itself and the

convex envelope of a concave function is the segment connecting $(l, f(l))$ and $(u, f(u))$. For functions of type (iii) and (iv) McCormick [McC76] and Maranas and Floudas [MF95] describe the construction of the convex envelopes. Using their ideas, we formally analyze certain well-known structural properties of these envelopes in order to determine C_{vex} and C_{poly}.

Observation 5.29 (convex-concave case)**.** *Let $g : [l, u] \subseteq \mathbf{R} \to \mathbf{R}$ be a twice continuously differentiable function and $w \in (l, u)$ such that $g''(x) > 0$ for all $x \in [l, w)$, $g''(w) = 0$, and $g''(x) < 0$ for all $x \in (w, u]$. Consider the equation*

$$g'(x) = \tfrac{g(u)-g(x)}{u-x}. \tag{5.5}$$

The following holds: (i) Let $x^* \in [l, u)$ be a solution of Equation (5.5)*, then $g''(x^*) > 0$. (ii) If Equation* (5.5) *exhibits a solution $x^* \in [l, u]$, then the solution is unique. (iii) Equation* (5.5) *possesses a solution $x^* \in [l, u)$ if and only if $g'(l) \leq \frac{g(u)-g(l)}{u-l}$. Moreover, if Equation* (5.5) *exhibits a solution $x^* \in [l, u]$, then*

$$\text{vex}_{[l,u]}[g](x) = \begin{cases} g(x), & x < x^*, \\ g'(x^*)\,(x - x^*) + g(x^*), & x \geq x^*. \end{cases}$$

If there is no $x^* \in [l, u)$ satisfying Equation (5.5)*, then* $\text{vex}_{[l,u]}[g](x) = \frac{g(u)-g(l)}{u-l}\,(x - l) + g(l)$.

The observation is illustrated in Figure 5.4. Given a convex-concave function g, the convex envelope is initially identical to the function over $[l, x^*]$, then it is given by the segment connecting $(x^*, g(x^*))$ and $(u, g(u))$. Moreover, the point x^* is the unique point such that the slope of the segment is equal to the slope of the tangent on g. In particular, g is strictly convex at x^*. Analogous results can be derived for concave-convex functions.

To bound the number of inflection points of $\alpha^\top f$ by one, we investigate the roots of $(\alpha^\top f)''(x) = \alpha_1 f_1''(x) + \alpha_2 f_2''(x) = 0$ over $x \in [l, u]$ which is equivalent to $\alpha_1 = -\alpha_2 f_2''(x)/f_1''(x)$ for strictly convex functions f_1 with $f_1''(x) > 0$ for all $x \in [l, u]$. There is at most one inflection point if $f_2''(x)/f_1''(x)$ is strictly monotone increasing, i.e., $(f_2''(x)/f_1''(x))' > 0$.

In our setting functions $g = \alpha^\top f$ are considered so that Equation (5.5) for convex-concave functions and the analogous equation for concave-

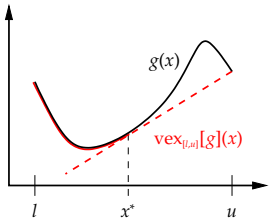


Figure 5.4.: A convex-concave function (black) and the parts of its convex envelope: The convex envelope (red) is the function itself until x^*, then it is given by the tangent on the function at x^*.

convex functions read

$$\alpha_1 f_1'(x) + \alpha_2 f_2'(x) = \frac{\big(\alpha_1 f_1(u)+\alpha_2 f_2(u)\big)-\big(\alpha_1 f_1(x)+\alpha_2 f_2(x)\big)}{u-x},$$
$$\alpha_1 f_1'(x) + \alpha_2 f_2'(x) = \frac{\big(\alpha_1 f_1(x)+\alpha_2 f_2(x)\big)-\big(\alpha_1 f_1(l)+\alpha_2 f_2(l)\big)}{x-l}.$$

Normalizing α to $\alpha_2 = -1$ in the first equation and $\alpha_2 = 1$ in the second equation, we can solve the resulting equations for α_1 and obtain $\alpha_1 = T_u(x)$ and $\alpha_1 = -T_l(x)$, respectively, where

$$T_l(x) := \begin{cases} f_2''(l)/f_1''(l), & x = l, \\ \frac{f_2(l)-f_2(x)-(l-x)f_2'(x)}{f_1(l)-f_1(x)-(l-x)f_1'(x)}, & x > l, \end{cases} \qquad T_u(x) := \begin{cases} \frac{f_2(u)-f_2(x)-(u-x)f_2'(x)}{f_1(u)-f_1(x)-(u-x)f_1'(x)}, & x < u, \\ f_2''(u)/f_1''(u), & x = u. \end{cases}$$

We exploit these terms, for instance, to show that $g = \alpha^\intercal f$ with $\alpha = (T_u(\bar{x}), -1)$ and $\bar{x} \in (l, u]$ is convex-concave and that the unique point x^* of Observation 5.29 and Figure 5.4 is identical to $\bar{x}$. Note that both the numerator and the denominator of the functions $T_l(x)$ and $T_u(x)$ are nonnegative if f_1 and f_2 are strictly convex since they express the underestimation of a convex function by its tangent, e.g., $f_1(l) \geq f_1(x) + (l - x)f_1'(x)$.

The previous considerations are now applied to determine the convexity patterns and the convex envelopes of functions $\alpha^\intercal f$. For this, it is sufficient to consider two cases of normalized α-vectors in order to deduce the convex envelope of $\alpha^\intercal f$ for all $\alpha \in \mathbf{R}^2$, namely $\alpha_2 = -1$ and $\alpha_2 = 1$.

Lemma 5.30 (Case 1: $\alpha_2 = -1$)**.** *Let $f : [l, u] \subseteq \mathbf{R} \to \mathbf{R}^2$ be a vector of three times continuously differentiable functions such that $f_i''(x) > 0$, $i = 1, 2$, and*

$(f_2''(x)/f_1''(x))' > 0$ *for all* $x \in [l,u]$. *Consider* $g^\alpha := \alpha_1 f_1 + \alpha_2 f_2$ *with* $\alpha_1 \in \mathbf{R}$, $\alpha_2 = -1$. *The following properties of* g^α *hold over* $[l,u]$*:*

(a) It is convex for all $\alpha_1 \geq f_2''(u)/f_1''(u)$, *concave for all* $\alpha_1 \leq f_2''(l)/f_1''(l)$, *and convex-concave for* $\alpha_1 \in (f_2''(l)/f_1''(l),\ f_2''(u)/f_1''(u))$.

(b) The term $T_u(x)$ *is continuous, strictly monotone increasing over* $[l,u]$, *and* $T_u(l) > f_2''(l)/f_1''(l)$. *For all* $\alpha_1 \in (T_u(l), T_u(u))$ *there is a unique* $\bar{x} \in (l,u)$ *with* $\alpha_1 = T_u(\bar{x})$ *such that*

$$\mathrm{vex}_{[l,u]}[g^\alpha](x) = \begin{cases} g^\alpha(x), & x \leq \bar{x}, \\ (g^\alpha)'(\bar{x})\,(x-\bar{x}) + g^\alpha(\bar{x}), & x > \bar{x}. \end{cases}$$

(c) For $\alpha_1 \leq T_u(l)$ *the convex envelope of* g^α *is vertex polyhedral.*

Proof. (a): Convexity is settled by the second partial derivative of g^α which reads $\alpha_1 f_1''(x) - f_2''(x)$. This expression is greater or equal to zero if and only if $\alpha_1 \geq f_2''(x)/f_1''(x)$. As $f_2''(x)/f_1''(x)$ is strictly increasing, the convexity/concavity characteristics follow.

(b): If $\alpha_1 \in (f_2''(l)/f_1''(l),\ f_2''(u)/f_1''(u))$, then g^α satisfies the assumptions of Observation 5.29 and Equation (5.5) evolves to

$$\alpha_1 f_1'(x) - f_2'(x) = \tfrac{(\alpha_1 f_1(u) - f_2(u)) - (\alpha_1 f_1(x) - f_2(x))}{u-x}$$

which is satisfied if and only if $\alpha_1 = T_u(x)$. Thus, for $\alpha_1 = T_u(\bar{x})$, $\bar{x} \in [l,u]$, we infer from Observation 5.29 that Equation (5.5) has a unique solution $x^* = \bar{x}$ and $(g^\alpha)''(\bar{x}) > 0$. In particular, if $\bar{x} = l$ and $\bar{\alpha} = (T_u(l), -1)$, then $(g^{\bar{\alpha}})''(l) > 0$ which implies that $g^{\bar{\alpha}}$ is not concave. Then, $T_u(l) = \bar{\alpha}_1 > f_2''(l)/f_1''(l)$ since g^α is concave for all $\alpha = (\alpha_1, -1)$ with $\alpha_1 \leq f_2''(l)/f_1''(l)$. The representation of the convex envelope follows from Observation 5.29.

To verify continuity of $T_u(x)$, it is sufficient to show that $T_u(x) \to f_2''(u)/f_1''(u)$ for $x \to u$ since all involved terms are continuous and $x = u$ is the only root of the denominator. We apply L'Hôpital's rule to determine the limit of $T_u(x)$ in u because the limit of both the numerator and denominator are zero if $x \to u$ (cf. [Wal04]). Then, $\lim_{x\to u} T_u(x) = [-(u-x)f_2''(u)]/[-(u-x)f_1''(u)] = f_2''(u)/f_1''(u)$. To show that $T_u(x)$ is strictly monotone increasing, we consider the first derivative of $T_u(x)$ and exploit that $(g^\alpha)''(x) > 0$ if $\alpha_1 = T_u(x)$. Let a and b denote the numerator and denominator of $T_u(x)$, respectively. Hence, $(T_u)'(x) = \frac{a'b - ab'}{b^2} > 0$ for all $x \in (l,u)$ if and only if $a'b - ab' = -(u-x)f_2''(x)b + a(u-x)f_1''(x) > 0$ for all $x \in (l,u)$. One can

check that $a, b > 0$ for all $x \in (l, u)$ because $f_i(x) + (u - x)f_i'(x)$ is the tangent on the strictly convex function f_i at x and thus less than $f_i(u)$. Therefore, $-(u - x)f_2''(x)b + a(u - x)f_1''(x) > 0$ is equivalent to $\frac{a}{b}f_1''(x) - f_2''(x) > 0$ and thus, equivalent to $(g^\alpha)''(x) > 0$.

(c): For $\alpha_1 \in [{}^{f_2''(l)}/_{f_1''(l)}\ ,\ T_u(l)]$ the convex envelope of g^α is given by its secant and thus, vertex polyhedral. For $\alpha_1 < {}^{f_2''(l)}/_{f_1''(l)}$ the function g^α is concave and its convex envelope is vertex polyhedral. □

In case of $\alpha_2 = 1$ analogous properties can be shown.

Lemma 5.31 (Case 2: $\alpha_2 = 1$)**.** *Let $f : [l, u] \subseteq \mathbf{R} \to \mathbf{R}^2$ be a vector of three times continuously differentiable functions such that $f_i''(x) > 0$, $i = 1, 2$, and $({}^{f_2''(x)}/_{f_1''(x)})' > 0$ for all $x \in [l, u]$. Consider $g^\alpha := \alpha_1 f_1 + \alpha_2 f_2$ with $\alpha_1 \in \mathbf{R}$, $\alpha_2 = 1$. The following properties of g^α hold over $[l, u]$:*

(a) It is convex for all $\alpha_1 \geq -{}^{f_2''(l)}/_{f_1''(l)}$, concave for all $\alpha_1 \leq -{}^{f_2''(u)}/_{f_1''(u)}$, and concave-convex for $\alpha_1 \in (-{}^{f_2''(u)}/_{f_1''(u)}\ ,\ -{}^{f_2''(l)}/_{f_1''(l)})$.

(b) The term $T_l(x)$ is continuous, strictly monotone increasing over $[l, u]$, and $T_l(u) < {}^{f_2''(u)}/_{f_1''(u)}$. For all $\alpha_1 \in (-T_l(u), -T_l(l))$ there is a unique $\bar{x} \in (l, u)$ with $\alpha_1 = -T_l(\bar{x})$ such that

$$\operatorname{vex}_{[l,u]}[g^\alpha](x) = \begin{cases} (g^\alpha)'(\bar{x})\,(x - \bar{x}) + g^\alpha(\bar{x}), & x < \bar{x}, \\ g^\alpha(x), & x \geq \bar{x}. \end{cases}$$

(c) For $\alpha_1 \leq -T_l(u)$ the convex envelope of g^α is vertex polyhedral.

The previous two lemmas lead to the characterization of C_{vex} and C_{poly}.

Theorem 5.32. *Let $f : [l, u] \subseteq \mathbf{R} \to \mathbf{R}^2$ be a vector of three times continuously differentiable functions such that $f_i''(x) > 0$, $i = 1, 2$, and $({}^{f_2''(x)}/_{f_1''(x)})' > 0$ for all $x \in [l, u]$. Then, $C_{\text{vex}} = \operatorname{cone}(\{\alpha^1_{\text{vex}}, \alpha^2_{\text{vex}}\})$ and $C_{\text{poly}} = \operatorname{cone}(\{\beta^1_{\text{poly}}, \beta^2_{\text{poly}}\})$, where*

$$\alpha^1_{\text{vex}} := \left(-\tfrac{f_2''(l)}{f_1''(l)}, 1\right), \quad \alpha^2_{\text{vex}} := \left(\tfrac{f_2''(u)}{f_1''(u)}, -1\right),$$
$$\beta^1_{\text{poly}} := (-T_l(u), 1), \quad \beta^2_{\text{poly}} := (T_u(l), -1).$$

Moreover,

$$Q_{[l,u]}[f] = \bigcap_{\alpha \in \bigcup_{i=1,2} \operatorname{cone}(\{\alpha^i_{\text{vex}}, \beta^i_{\text{poly}}\})} \{(x, z_1, z_2) \in \mathbf{R}^3 \mid \alpha^\intercal z \geq \operatorname{vex}_{[l,u]}[\alpha^\intercal f](x)\}.$$

Proof. The theorem follows from Lemmas 5.26, 5.30, and 5.31. □

In Figure 5.5 we illustrate Theorem 5.32. The figure displays the subdivision of $\alpha \in \mathbf{R}^2$ with respect to the convex envelope of $\alpha^\top f$ over a domain $[l, u]$. For $\alpha \in C_{\text{vex}}$ the function $\alpha^\top f$ is convex and thus, it is identical to its convex envelope. For $\alpha \in C_{\text{poly}}$ the convex envelope of $\alpha^\top f$ is vertex polyhedral.

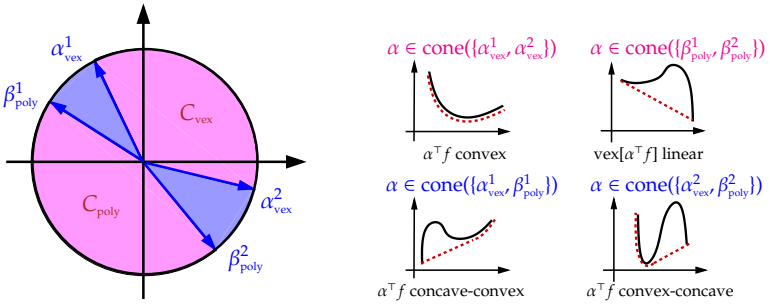


Figure 5.5.: Subdivision for $\alpha \in \mathbf{R}^2$ w.r.t. the type of the convex envelope of $\alpha^\top f$. An example from each subdivision is given in the figures at the right hand side, where the functions $\alpha^\top f$ (black) and their convex envelopes (red) are displayed.

Recall that a central question of this chapter is to identify subsets of $\alpha \in \mathbf{R}^m$ such that the corresponding constraints $\text{vex}_{[l,u]}[\alpha^\top f](x) \leq \alpha^\top z$ are necessary and sufficient to describe $Q_{[l,u]}[f]$. Due to Theorem 5.32 the points $\alpha \in \text{int}(C_{\text{vex}}) \cup \text{int}(C_{\text{poly}})$ are not necessary for this. After exclusion of the unnecessary constraints it follows obviously that the constraints corresponding to $\alpha \in \mathbf{R}^m \setminus (\text{int}(C_{\text{vex}}) \cup \text{int}(C_{\text{poly}}))$ are sufficient for $Q_{[l,u]}[f]$. We show that, up to scaling, none of these constraints can be obtained by a conic combination of other constraints. Within this process we further derive a complete outer description of $Q_{[l,u]}[f]$ by supporting hyperplanes.

Theorem 5.33. *Let $f : [l, u] \subseteq \mathbf{R} \to \mathbf{R}^2$ be a vector of three times continuously differentiable functions such that $f_i''(x) > 0$, $i = 1, 2$, and $(f_2''(x)/f_1''(x))' > 0$ for all $x \in [l, u]$. The vectors α^i_{vex} and β^i_{poly}, $i = 1, 2$, are defined according to Theorem 5.32. Then, up to scaling, none of the constraints $\alpha^\top z \geq \text{vex}_{[l,u]}[\alpha^\top f](x)$*

with $\alpha \in \operatorname{int}(\operatorname{cone}(\{\alpha^i_{\text{vex}}, \beta^i_{\text{poly}}\}))$, $i = 1, 2$, *can be represented as conic combination of other constraints* $(\alpha^i)^\top z \geq \operatorname{vex}_{[l,u]}[(\alpha^i)^\top f](x)$, $\alpha^i \in \mathbf{R}^m$. *Moreover,*

$$Q_{[l,u]}[f] = \bigcap_{\bar{x}\in[l,u]} \left\{ (x, z_1, z_2) \;\middle|\; \begin{array}{l} \alpha^\top z \geq (\alpha^\top f)'(\bar{x})(x - \bar{x}) + \alpha^\top f(\bar{x}), \\ \alpha^\top z \geq (\alpha^\top f)'(\bar{x})(x - \bar{x}) + \alpha^\top f(\bar{x}), \\ \textit{where } \alpha \in \{(T_u(\bar{x}), -1), (-T_l(\bar{x}), 1)\} \end{array} \right\}. \quad (5.6)$$

We first need an auxiliary result.

Lemma 5.34. *Let* $f : [l, u] \subseteq \mathbf{R} \to \mathbf{R}^2$ *be a vector of three times continuously differentiable functions such that* $f_i''(x) > 0$, $i = 1, 2$, *and* $(f_2''(x)/f_1''(x))' > 0$ *for all* $x \in [l, u]$. *Then,* $T_l(u) > T_u(l)$ *and* $T_u(x) > T_l(x)$ *for all* $x \in [l, u]$.

Proof. Let $\alpha = (T_u(l), -1)$. The function $\alpha^\top f$ is convex-concave and the convex envelope $\operatorname{vex}_{[l,u]}[\alpha^\top f]$ is vertex polyhedral according to Lemma 5.30. The function $(-\alpha)^\top f$ is concave-convex. If the convex envelope of $(-\alpha)^\top f$ is also vertex polyhedral, it follows from

$$\operatorname{vex}_{[l,u]}[(-\alpha)^\top f] = \tfrac{(-\alpha)^\top f(u) - (-\alpha)^\top f(l)}{u-l}(x - l) + (-\alpha)^\top f(l) = -\operatorname{vex}_{[l,u]}[\alpha^\top f]$$

and $-\operatorname{vex}_{[l,u]}[(-\alpha)^\top f] = \operatorname{cave}_{[l,u]}[\alpha^\top f]$ that $\operatorname{vex}_{[l,u]}[\alpha^\top f] = \operatorname{cave}_{[l,u]}[\alpha^\top f] = \alpha^\top f$. As $\operatorname{vex}_{[l,u]}[\alpha^\top f]$ is an affine function, the same holds for $\alpha^\top f$. This contradicts the fact that $\alpha^\top f$ is convex-concave and thus, $\operatorname{vex}_{[l,u]}[(-\alpha)^\top f]$ is not vertex polyhedral. Lemma 5.31 (c) then yields that $-T_u(l) = -\alpha_1 > -T_l(u)$ so that we proved the first claim $T_u(l) < T_l(u)$.

Lemmas 5.30 (b) and 5.31 (b) state further that $T_u(l) > f_2''(l)/f_1''(l) = T_l(l)$ and $T_u(u) = f_2''(u)/f_1''(u) > T_l(u)$ so that $T_l(l) < T_u(l) < T_l(u) < T_u(u)$. As $T_l(x)$ and $T_l(x)$ are strictly monotone increasing, there are $x^1, x^2 \in [l, u]$ such that $T_u(x^1) = T_l(x^2)$. Then, $(\alpha^1)^\top f(x)$ with $\alpha^1 = (T_u(x^1), -1)$ is convex-concave and strictly convex at $x = x^1$ (see Observation 5.29 (i) and Lemma 5.30) while $(\alpha^2)^\top f(x)$ with $\alpha^2 = (-T_l(x^2), 1)$ is concave-convex and strictly convex at $x = x^2$. As $(\alpha^1)^\top f$ is convex-concave and $\alpha^1 = -\alpha^2$ it follows that $x^1 < x^2$. Moreover, $T_u(x)$ is strictly monotone increasing which implies that $T_u(x^2) > T_u(x^1) = T_l(x^2)$ and shows that $T_u(x) > T_l(x)$ for all $x \in [l, u]$. □

We are now ready to prove Theorem 5.33.

Proof of Theorem 5.33. In order to show that none of the constraints $\bar{\alpha}^\top z \geq \operatorname{vex}_{[l,u]}[\bar{\alpha}^\top f](x)$ with $\bar{\alpha} \in \operatorname{int}(\operatorname{cone}(\{\alpha^i_{\text{vex}}, \beta^i_{\text{poly}}\}))$, $i = 1, 2$, is a conic combi-

nation of other constraints of this form, we write $\bar{\alpha}$ as $\bar{\alpha} = (T_u(\bar{x}), -1)$ or $\bar{\alpha} = (-T_l(\bar{x}), 1)$ for some $\bar{x} \in (l, u)$ and consider the constraints at $x = \bar{x}$, where $\text{vex}_{[l,u]}[\bar{\alpha}^\top f](\bar{x}) = \bar{\alpha}^\top f(\bar{x})$ (Lemmas 5.30 (b) and 5.31 (b)). Let $\bar{\alpha} = (T_u(\bar{x}), -1)$ and assume that there are some $\mu_i \geq 0$, $\alpha^i \in \mathbf{R}^m$, $\alpha^i \neq \bar{\alpha}$, with $\bar{\alpha} = \sum_i \mu_i \alpha^i$ and $\text{vex}_D[\bar{\alpha}^\top f](\bar{x}) = \sum_i \mu_i \, \text{vex}_D[(\alpha^i)^\top f](\bar{x})$. As $\text{vex}_D[\bar{\alpha}^\top f](\bar{x}) = \bar{\alpha}^\top f(\bar{x})$ and $\text{vex}_D[(\alpha^i)^\top f](\bar{x}) \leq (\alpha^i)^\top f(\bar{x})$, it has to hold that $\text{vex}_D[(\alpha^i)^\top f](\bar{x}) = (\alpha^i)^\top f(\bar{x})$. We show that all $\alpha^i \in \mathbf{R}^m$, $\alpha^i \neq \bar{\alpha}$, with $\text{vex}_D[(\alpha^i)^\top f](\bar{x}) = (\alpha^i)^\top f(\bar{x})$ are contained in the open halfspace $H_{\bar{\alpha}} := \{\alpha \in \mathbf{R}^2 \mid -\bar{\alpha}_2\alpha_1 + \bar{\alpha}_1\alpha_2 = \alpha_1 + T_u(\bar{x})\alpha_2 > 0\}$ which does not contain $\bar{\alpha}$. This contradicts the existence of $\mu_i \geq 0$ with $\bar{\alpha} = \sum_i \mu_i \alpha^i$ because $\bar{\alpha} \notin H_{\bar{\alpha}}$.

If $\alpha^i_2 = -1$ and $\alpha^i \neq \bar{\alpha}$, then $\text{vex}_D[(\alpha^i)^\top f](\bar{x}) = (\alpha^i)^\top f(\bar{x})$ if and only if $\alpha^i_1 > T_u(\bar{x})$ (see Lemma 5.30) and thus, $-\bar{\alpha}_2\alpha^i_1 + \bar{\alpha}_1\alpha^i_2 = \alpha^i_1 + T_u(\bar{x})(-1) > T_u(\bar{x}) - T_u(\bar{x}) = 0$. This means $\alpha^i \in H_{\bar{\alpha}}$. If $\alpha^i_2 = 0$, then $\text{vex}_D[(\alpha^i)^\top f](\bar{x}) = (\alpha^i)^\top f(\bar{x}) = \alpha^i_1 f_1(\bar{x})$ if and only if $\alpha^i_1 \geq 0$. Note that we can exclude $\alpha^i = (0, 0)$ from our considerations as the corresponding constraint $(\alpha^i)^\top z \geq \text{vex}_{[l,u]}[(\alpha^i)^\top f](x)$ yields $0 \geq 0$ which is useless for our purposes. Thus, the interesting α^i satisfy $\alpha^i_1 > 0$ and are contained in H_α. If $\alpha^i_2 = 1$, then $\text{vex}_D[(\alpha^i)^\top f](\bar{x}) = (\alpha^i)^\top f(\bar{x})$ if and only if $\alpha^i_1 \geq -T_l(\bar{x})$. We can infer from Lemma 5.34 that $-T_u(\bar{x}) < -T_l(\bar{x})$ and thus, $-\bar{\alpha}_2\alpha^i_1 + \bar{\alpha}_1\alpha^i_2 \geq (-T_l(\bar{x})) + T_u(\bar{x})(1) > -T_u(\bar{x}) + T_u(\bar{x}) = 0$ which implies $\alpha^i \in H_{\bar{\alpha}}$. Thus, all $\alpha^i \in \mathbf{R}^m$, $\alpha^i \neq \bar{\alpha}$, with $\text{vex}_D[(\alpha^i)^\top f](\bar{x}) = (\alpha^i)^\top f(\bar{x})$ are contained in the open halfspace $H_{\bar{\alpha}}$. This leads to a contradiction. An analogous procedure can be applied for $\bar{\alpha} = (-T_l(\bar{x}), 1)$ showing that the constraint $\bar{\alpha}^\top z \geq \text{vex}_{[l,u]}[\bar{\alpha}^\top f](x)$ is no surrogate of other constraints.

Next we prove the linear inequality description of $Q_{[l,u]}[f]$. Theorem 5.32 yields

$$Q_{[l,u]}[f] = \bigcap_{\bar{x} \in [l,u]} \left\{ (x, z_1, z_2) \;\middle|\; \begin{array}{ll} \alpha^\top z \geq \text{vex}_{[l,u]}[\alpha^\top f](x) & \text{with } \alpha = (T_u(\bar{x}), -1), \\ \alpha^\top z \geq \text{vex}_{[l,u]}[\alpha^\top f](x) & \text{with } \alpha = (-T_l(\bar{x}), 1) \end{array} \right\}.$$

Let $\bar{\alpha} = (T_u(\bar{x}), -1)$ for an $\bar{x} \in [l, u]$, then $\bar{\alpha}^\top f$ is convex-concave and its convex envelope is piecewise defined by $\text{vex}_{[l,u]}[\bar{\alpha}^\top f](x) = \alpha^\top f$ for $x \in [l, \bar{x})$ and by $\text{vex}_{[l,u]}[\bar{\alpha}^\top f](x) = (\bar{\alpha}^\top f)'(\bar{x})(x - \bar{x}) + \bar{\alpha}^\top f(\bar{x})$ for $x \in [\bar{x}, u]$. Thus, the linear inequality description in Equation (5.6) follows over this domain $[\bar{x}, u]$. For $x \in [l, \bar{x})$ we show that $\bar{\alpha}^\top z \geq \text{vex}_{[l,u]}[\bar{\alpha}^\top f](x) = \bar{\alpha}^\top f(x)$ is dominated by the family of constraints $\beta^\top z \geq (\beta^\top f)'(\bar{x})(x - \bar{x}) + \bar{\alpha}^\top f(\bar{x})$ with $\beta_1 = T_u(y) < T_u(\bar{x})$, i.e., $y \in [l, \bar{x})$ and $\beta_2 = -1$. Then, the result follows.

To show the claim, consider the constraint $\bar{\alpha}^\top z \geq \text{vex}_{[l,u]}[\bar{\alpha}^\top f](x)$ at $x = y \in [l, \bar{x})$, i.e., $T_u(\bar{x})z_1 - z_2 \geq \text{vex}_{[l,u]}[\bar{\alpha}^\top f](y) = T_u(\bar{x})f_1(y) - f_2(y)$. We compare this constraint with the hyperplane constraint corresponding to

$\beta = (T_u(y), -1)$ which is given by $T_u(y)z_1 - z_2 \geq (\beta^\intercal f)'(y)(x-y) + \beta^\intercal f(y)$. By the choice of y and $\beta = (T_u(y), -1)$ it follows that $\mathrm{vex}_{[l,u]}[\beta^\intercal f](y) = \beta^\intercal f(y) = T_u(y)f_1(y) - f_2(y) = (\beta^\intercal f)'(y)(x-y) + \beta^\intercal f(y)$. Thus, the hyperplane constraint at $x = y$ is equivalent to $T_u(y)z_1 - z_2 \geq T_u(y)f_1(y) - f_2(y)$. Both the $\bar{\alpha}$- and the β-constraint give upper bounds on z_2:

$$\begin{aligned} \bar{\alpha}: &\quad z_2 \leq T_u(\bar{x})\, z_1 - T_u(\bar{x})f_1(y) + f_2(y), \\ \beta: &\quad z_2 \leq T_u(y)\, z_1 - T_u(y)f_1(y) + f_2(y). \end{aligned}$$

The β-constraint is more restrictive because $T_u(\bar{x})z_1 - T_u(\bar{x})f_1(y) + f_2(y) \geq T_u(y)z_1 - T_u(y)f_1(y) + f_2(y)$ is equivalent to $(T_u(\bar{x}) - T_u(y))z_1 \geq (T_u(\bar{x}) - T_u(y))f_1(y)$ which is true since $T_u(x)$ is strictly monotone increasing, $\bar{x} > y$, and $z_1 \geq f_1(y)$ is implied by the valid constraints of the individual convex hull of the graph of f_1 given by $Q_{[l,u]}[f_1] = \{(x, z_1) \mid f_1(x) = \mathrm{vex}_{[l,u]}[f_1] \leq z_1 \leq \mathrm{cave}_{[l,u]}[f_1], x \in [l,u]\}$. Thus, for $\bar{\alpha} = (T_u(\bar{x}), -1)$ we showed that the constraint $\bar{\alpha}^\intercal z \geq \mathrm{vex}_{[l,u]}[\bar{\alpha}^\intercal f](x)$ is induced by other constraints for $x < \bar{x}$.

The claim can be proven analogously for $\bar{\alpha} = (-T_l(\bar{x}), 1)$ with $\bar{x} \in [l,u]$. Thus, the result follows. □

In Theorems 5.32 and 5.33 the simultaneous convex hull $Q_{[l,u]}[f]$ is represented as the intersection of uncountably many constraints corresponding to the supporting hyperplanes on $Q_{[l,u]}[f]$. However, $Q_{[l,u]}[f] \subseteq \mathbf{R}^3$ is a nonempty closed set and $\mathbf{R}^3$ is a separable Banach space and thus, $Q_{[l,u]}[f]$ can be described by a countable subset of constraints (cf. [AB06]). Hence, we can give a final answer to the question regarding subsets of $\alpha \in \mathbf{R}^m$ such that the description of $Q_{[l,u]}[f]$ via the constraints $\mathrm{vex}_{[l,u]}[\alpha^\intercal f](x) \leq \alpha^\intercal z$ is necessary and sufficient. We infer from Theorems 5.32 and 5.33 that there is not necessarily a unique set satisfying this, but each of these sets is a countable subset of $\mathbf{R}^m \setminus (\mathrm{int}(C_{\mathrm{vex}}) \cup \mathrm{int}(C_{\mathrm{poly}}))$. From a computational point of view, the representation of $Q_{[l,u]}[f]$ via countably infinitely many constraints is not applicable. To overcome this problem, we devote the remainder of this section to the construction of a strong, basic relaxation of $Q_{[l,u]}[f]$, and then provide a separation result such that constraints can be added to the basic relaxation to cut off any point $(x, z) \notin Q_{[l,u]}[f]$.

A Basic Relaxation We propose a relaxation of $Q_{[l,u]}[f]$ based on the extreme rays $\alpha^1_{\mathrm{vex}}, \alpha^2_{\mathrm{vex}}$ of C_{vex} and $\beta^1_{\mathrm{vex}}, \beta^2_{\mathrm{vex}}$ of C_{poly} defined in Theorem 5.32. The advantage of this choice is that the computation of the corresponding convex envelopes is easy and that the resulting constraints induce all

other constraints $\text{vex}_{[l,u]}[\alpha^\top f](x) \leq \alpha^\top z$ with $\alpha \in \text{int}(C_{\text{vex}}) \cup \text{int}(C_{\text{poly}})$. The strength of this relaxation is illustrated in the next example.

Example 5.35 (Example 5.5 continued). Let $f := (x^2, x^3)$ and $[l, u] = [1, 2]$. The standard relaxation is given by the individual convex and concave envelopes

$$\begin{aligned} R_{\text{Std}} &= \{(x, z_1, z_2) \mid \text{vex}_D[f_i](x) \leq z_i,\ \text{vex}_D[-f_i](x) \leq -z_i,\ i = 1, 2\} \\ &= \{(x, z_1, z_2) \mid f_1(x) \leq z_1,\ -3x + 2 \leq -z_1,\ f_2(x) \leq z_2,\ -7x + 6 \leq -z_2\}. \end{aligned}$$

The constraints of R_{Std} correspond to (α_1, α_2)-values of $(1, 0), (-1, 0), (0, 1)$, and $(0, -1)$, which are interior points of the cones C_{vex} and C_{poly} generated by the extreme rays $\alpha^1_{\text{vex}} = (6, -1)$, $\alpha^2_{\text{vex}} = (-3, 1)$, and $\beta^1_{\text{poly}} = (4, -1)$, $\beta^2_{\text{poly}} = (-5, 1)$, respectively. Hence, the constraints of R_{Std} are implied by the constraints corresponding to these extreme rays. Our proposed basic relaxation is given by

$$\begin{aligned} R_{\text{Bsc}} &= \left\{(x, z_1, z_2) \;\middle|\; \begin{array}{l} \text{vex}_{[1,2]}[(\alpha^i_{\text{vex}})^\top f](x) \leq (\alpha^i_{\text{vex}})^\top z,\ i = 1, 2 \\ \text{vex}_{[1,2]}[(\beta^i_{\text{poly}})^\top f](x) \leq (\beta^i_{\text{poly}})^\top z,\ i = 1, 2 \end{array} \right\} \\ &= \left\{(x, z_1, z_2) \;\middle|\; \begin{array}{rrrr} 6x^2 - x^3 \leq & 6z_1 - z_2, & 5x - 2 \leq & 4z_1 - z_2 \\ -3x^2 + x^3 \leq & -3z_1 + z_2, & -8x + 4 \leq & -5z_1 + z_2 \end{array} \right\}. \end{aligned}$$

This relaxation in contained in the linear relaxation $S^3_{[1,2]}$ of $Q_{[1,2]}[f]$ defined in Example 5.5, which is given by

$$S^3_{[1,2]} = \left\{(x, z_1, z_2) \;\middle|\; \begin{array}{rrrr} 12x - 8 \leq & 6z_1 - z_2, & 5x - 2 \leq & 4z_1 - z_2 \\ -3x + 1 \leq & -3z_1 + z_2, & -8x + 4 \leq & -5z_1 + z_2 \end{array} \right\}.$$

Note that the (α_1, α_2)-values of these constraints are identical to ones used in R_{Bsc}. In particular, two constraints of R_{Bsc} and $S^3_{[1,2]}$ are identical, namely the linear ones corresponding to β^1_{poly} and β^2_{poly}. Further, for the remaining constraints it holds that $12x - 8$ is the tangent on the convex term $6x^2 - x^3$ at $x = u = 2$ and $-3x + 1$ is the tangent on the convex term $-3x^2 + x^2$ at $x = l = 1$, i.e.,

$$12x - 8 \leq 6x^2 - x^3 \leq 6z_1 - z_2 \quad \text{and} \quad -3x + 1 \leq -3x^2 + x^2 \leq -3z_1 + z_2$$

for all $x \in [1, 2]$. Thus, we conclude that the proposed basic relaxation can be seen as an extension of the linear relaxation $S^3_{[1,2]}$ for the moment curve to more general vectors of two functions.

The quality of the different convex relaxations is compared in Table 5.2, where the volume of each relaxation is computed by numerical integration with Mathematica 8 [Wol08]. The numbers show an enormous difference between the standard relaxation R_{Std} and the suggested relaxation R_{Bsc} based on Theorem 5.32. While the volume of R_{Bsc} and $Q_{[1,2]}[f]$ differ by a factor of two, the factor for the difference between R_{Std} and $Q_{[1,2]}[f]$ is 27. The difference between the linear relaxation $S^3_{[1,2]}$ and the (strictly) convex relaxation R_{Bsc} is reasonable and accounts for a factor of 1.5. ◇

	R_{Std}	$S^3_{[1,2]}$	R_{Bsc}	$Q_{[1,2]}[(x^2, x^3)]$
Volume	0.1500	0.0185	0.0119	0.0055

Table 5.2.: Volumes of the different convex relaxations for $Q_{[1,2]}[(x^2, x^3)]$.

In the example, the constraints of the standard relaxation R_{Std} are induced by the constraints of the basic relaxation R_{Bsc} so that $R_{\text{Std}} \subseteq R_{\text{Bsc}}$. This is generally true as the (α_1, α_2)-values $(1, 0), (-1, 0), (0, 1)$, and $(0, -1)$ corresponding to R_{Std} are always interior points of $C_{\text{vex}} = \text{cone}\{\alpha^1_{\text{vex}}, \alpha^2_{\text{vex}}\}$ and $C_{\text{poly}} = \text{cone}\{\beta^1_{\text{poly}}, \beta^2_{\text{poly}}\}$, i.e., $R^2_{\leq 0} \subsetneq C_{\text{poly}}$ and $R^2_{\geq 0} \subsetneq C_{\text{vex}}$. This can be easily verified by using the assumptions and definitions in Theorem 5.32.

A Separation Result The proposed basic relaxation can be seen as an initial relaxation of $Q_D[f]$ for which we further deduce linear inequalities that separate any $(x, z) \notin Q_D[f]$ from $Q_D[f]$. Such a procedure is best suited for the concept of branch-and-bound algorithms which usually start with an initial relaxation at each node of the branching tree, solve this relaxation, and then check if additional constraints can be added to the relaxation in order to cut off the current solution.

To illustrate the potential of our approach and to explain some notation, we compare the simultaneous convex hull $Q_{[l,u]}[f]$ to the relaxation obtained by the individual convex and concave envelopes for f_1 and f_2 in Figure 5.6. The green colored area represents the projection of $Q_{[l,u]}[f]$ onto the (z_1, z_2)-space at a fixed $\bar{x} \in (l, u)$ while the individual relaxations correspond to the dashed box. Usually a closed-form description of $Q_D[f]$ is not known so that a strong relaxation of this set is needed, e.g., the proposed basic relaxation. Next, we refine this relaxation by deriving a supporting hyperplane from Theorem 5.33 for each

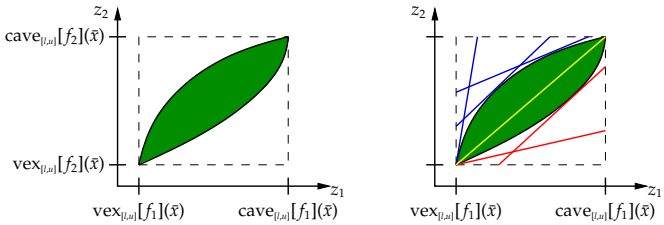


Figure 5.6.: The figure displays the projection of $Q_D[f]$ onto the (z_1, z_2)-space for a fixed $\bar{x} \in D$ and indicates its relaxation by supporting hyperplanes.

point $(\bar{x}, \bar{z}) \notin Q_D[f]$ that cuts off $(\bar{x}, \bar{z})$. For this, consider Figure 5.6 (b), where the red hyperplanes correspond to constraints $\alpha^\intercal z \geq a^\intercal x + \gamma$ with $\alpha_2 = 1$ and hence $\alpha \in \text{cone}(\{\alpha^1_{\text{vex}}, \beta^1_{\text{poly}}\})$ while the blue hyperplanes belong to constraints $\alpha^\intercal z \geq a^\intercal x + \gamma$, where $\alpha_2 = -1$ and $\alpha \in \text{cone}(\{\alpha^2_{\text{vex}}, \beta^2_{\text{poly}}\})$. If the point $(\bar{x}, \bar{z})$, which we aim to cut off, lies above the yellow segment in Figure 5.6 (b) connecting $(z_1, z_2) = (\text{vex}_{[l,u]}[f_1](\bar{x}), \text{vex}_{[l,u]}[f_2](\bar{x}))$ and $(z_1, z_2) = (\text{cave}_{[l,u]}[f_1](\bar{x}), \text{cave}_{[l,u]}[f_2](\bar{x}))$, it can be cut off by the blue hyperplanes. Otherwise, the point is separated from $Q_D[f]$ by the red hyperplanes. Formally, this means if

$$\bar{z}_2 \geq \frac{\text{cave}_{[l,u]}[f_2](\bar{x}) - \text{vex}_{[l,u]}[f_2](\bar{x})}{\text{cave}_{[l,u]}[f_1](\bar{x}) - \text{vex}_{[l,u]}[f_1](\bar{x})} (\bar{z}_1 - \text{vex}_{[l,u]}[f_1](\bar{x})) + \text{vex}_{[l,u]}[f_2](\bar{x}), \tag{5.7}$$

then $(\bar{x}, \bar{z})$ is cut off by a hyperplane $\alpha^\intercal z \geq (\alpha^\intercal f)'(y)(x - y) + \alpha^\intercal f(y)$ with $\alpha = (T_u(y), -1)$, $y \in [l, u]$, and otherwise, with $\alpha = (-T_l(y), 1)$, $y \in [l, u]$. We show that the determination of a separating hyperplane is equivalent to minimizing a certain function over $[l, u]$ which is first strictly decreasing and then strictly increasing so that the point satisfying the first order necessary condition is the global optimum. Such functions are called *unimodal*.

Lemma 5.36 ($\alpha_2 = -1$)**.** *Let $f : [l, u] \subseteq \mathbf{R} \to \mathbf{R}^2$ be a vector of three times continuously differentiable functions such that $f_i''(x) > 0$, $i = 1, 2$, and $(f_2''(x)/f_1''(x))' > 0$ for all $x \in [l, u]$. Assume that $(\bar{x}, \bar{z}_1, \bar{z}_2) \notin Q_D[f]$, $(\bar{x}, \bar{z}_1) \in Q_D[f_1]$, and Equation (5.7) is satisfied. If $\bar{x} = u$, then $(\bar{x}, \bar{z}_1, \bar{z}_2)$ can be cut off by $\alpha^\intercal z \geq (\alpha^\intercal f)'(y)(x - y) + \alpha^\intercal f(y)$ with $\alpha = (T_u(y), -1)$ for any $y \in [l, u]$. If $\bar{x} \in [l, u)$, then $(\bar{x}, \bar{z}_1, \bar{z}_2)$ can be cut off by the hyperplane*

$\alpha^\intercal z \geq (\alpha^\intercal f)'(y)(x-y) + \alpha^\intercal f(y)$ *with* $\alpha = (T_u(y), -1)$*, where* y *is the unique minimizer of an unimodal function over* $[l, \bar{x}]$*, i.e.,*

$$\{y\} = \operatorname{argmin}\left\{T_u(y)\,\bar{z}_1 - (\alpha^\intercal f)'(y)(\bar{x}-y) - \alpha^\intercal f(y) \mid y \in [l,\bar{x}]\right\}.$$

The minimizer y *is determined as the unique solution of the system*

$$(\bar{x}-u)\,f_1(y) + (f_1(u)-\bar{z}_1)\,y + u\,\bar{z}_1 - \bar{x}\,f_1(u) = 0 \quad \textit{and} \quad y \leq \bar{x}. \tag{5.8}$$

Proof. Theorem 5.33 yields that linear constraints of the form $\alpha^\intercal z \geq (\alpha^\intercal f)'(y)(x-y) + \alpha^\intercal f(y)$ are sufficient to describe $Q_{[l,u]}[f]$. If $(\bar{x}, \bar{z}_1, \bar{z}_2)$ satisfies Equation (5.7), we are in the case of the blue hyperplanes in Figure 5.6 (b), i.e., $\alpha = (T_u(y), -1)$ with $y \in [l,u]$. The maximal z_2 for the given $(\bar{x}, \bar{z}_1)$ such that $(\bar{x}, \bar{z}_1, z_2) \in Q_{[l,u]}[f]$ is determined over all constraints $T_u(y)\bar{z}_1 - z_2 \geq (\alpha^\intercal f)'(y)(\bar{x}-y) + \alpha^\intercal f(y)$ and thus,

$$z_2 \leq \min_{y\in[l,u]} \{\underbrace{T_u(y)\bar{z}_1 - (\alpha^\intercal f)'(y)(\bar{x}-y) - \alpha^\intercal f(y)}_{=:h(y)}\}.$$

We show that $h(y)$ is first strictly monotone decreasing and then strictly monotone increasing. Let a and b denote the numerator and denominator of T_u, i.e., $T_u = {}^a\!/_b$. The first derivative of $h(y)$ reads $h'(y) = \frac{(\alpha^\intercal f)''(y)\; t(y)}{b}$, where $t(y) := (\bar{x}-u)\,f_1(y) + (f_1(u)-\bar{z}_1)\,y + u\,\bar{z}_1 - f_1(u)\,\bar{x}$ (cf. Equation (5.8)). The sign of $h'(y)$ is determined by $t(y)$ because b is nonnegative and $(\alpha^\intercal f)''(y) > 0$ for all $\bar{x} \in [l,u)$ (see Observation 5.29 (i)).

If $\bar{x} = u$, then $\bar{z}_1 = f_1(u)$ is implied by $(\bar{x}, \bar{z}_1) \in Q_{[l,u]}[f_1]$. In this case $h(y) = f_2(u)$ and $h'(y) = 0$ for all $y \in [l,u]$. If $\bar{x} \in [l,u)$, then $t(y)$ is strictly concave since $\bar{x} - u < 0$ and f_1 is strictly convex. Thus, there are at most two roots for $t(y) = 0$. One root is attained at $y = u$. If $t(l) \leq 0$, the second root is attained in $[l,u)$ due to concavity of t. The expression $t(l) \leq 0$ is equivalent to $(u-l)\,\bar{z}_1 \leq (f_1(u) - f_1(l))\,\bar{x} + (u-l)\,f_1(l)$ and thus, equivalent to $\bar{z}_1 \leq \frac{f_1(u)-f_1(l)}{u-l}\,(\bar{x}-l) + f_1(l)$. The latter expression is true since we assume $(\bar{x}, \bar{z}_1) \in Q_{[l,u]}[f_1] = \{(x,z) \mid f_1(x) \leq z_1 \leq \frac{f_1(u)-f_1(l)}{u-l}\,(x-l) + f_1(l)\}$. Therefore, the case $t(l) > 0$ cannot occur and there is a $y^* \in [l,u)$ with $h'(y) < 0$ for all $y \in [l, y^*)$, $h'(y^*) = 0$, and $h'(y) > 0$ for all $y \in (y^*, u]$ so that the objective function h is unimodal. Moreover, the optimal solution is the unique solution of $t(y) = 0$ which is stated in Equation (5.8).

It remains to show that $h'(y^*) = t(y^*) = 0$ is satisfied for $y^* \in [l, \bar{x}]$. Recall that $t(y)$ is strictly concave with at most two roots and one root is attained at $y = u$. We observe that $t(\bar{x}) = (u - \bar{x})(\bar{z}_1 - f_1(\bar{x})) \geq 0$ since $\bar{x} \in [l, u)$ and $(\bar{x}, \bar{z}_1) \in \mathcal{Q}_{[l,u]}[f_1]$. Therefore, $t(y) > 0$ for all $y \in (\bar{x}, u)$ so that no minimum is attained over this domain. □

Lemma 5.37 ($\alpha_2 = 1$)**.** *Let $f : [l, u] \subseteq \mathbf{R} \to \mathbf{R}^2$ be a vector of three times continuously differentiable functions such that $f_i''(x) > 0$, $i = 1, 2$, and $(f_2''(x)/f_1''(x))' > 0$ for all $x \in [l, u]$. Assume that $(\bar{x}, \bar{z}_1, \bar{z}_2) \notin \mathcal{Q}_D[f]$, $(\bar{x}, \bar{z}_1) \in \mathcal{Q}_D[f_1]$, and Equation (5.7) is not satisfied. If $\bar{x} = l$, then $(\bar{x}, \bar{z}_1, \bar{z}_2)$ can be cut off by $\alpha^\intercal z \geq (\alpha^\intercal f)'(y)(x - y) + \alpha^\intercal f(y)$ with $\alpha = (-T_l(y), 1)$ for any $y \in [l, u]$. If $\bar{x} \in (l, u]$, then $(\bar{x}, \bar{z}_1, \bar{z}_2)$ can be cut off by the hyperplane $\alpha^\intercal z \geq (\alpha^\intercal f)'(y)(x - y) + \alpha^\intercal f(y)$ with $\alpha = (-T_l(y), 1)$, where y is the unique maximizer of*

$$\max \{T_l(y)\, \bar{z}_1 + (\alpha^\intercal f)'(y)(\bar{x} - y) + \alpha^\intercal f(y) \mid y \in [\bar{x}, u]\}.$$

The maximizer y is determined as the unique solution of the system

$$(l - \bar{x})\, f_1(y) + (z_1 - f_1(u))\, y + l\, z_1 + \bar{x}\, f_1(l) = 0 \quad \text{and} \quad y \in [\bar{x}, u]. \tag{5.9}$$

Remark 5.38. Note that $R_{\text{Bsc}} \subseteq R_{\text{Std}} = \{(x, z_1, z_2 \mid (x, z_1) \in \mathcal{Q}_D[(f_1)], (x, z_2) \in \mathcal{Q}_D[(f_2)]\}$ so that the assumption $(x, z_1) \in \mathcal{Q}_D[(f_1)]$ in Lemmas 5.36 and 5.37 can be replaced by the stricter assumption $(x, z_1, z_2) \in R_{\text{Bsc}}$. This allows to start with the proposed basic relaxation R_{Bsc} and then to cut off any $(x, z) \notin \mathcal{Q}_D[f]$.

The next example shows that the separation problem can be solved analytically for some classes of functions leading to closed-form expressions for $\mathcal{Q}_D[f]$.

Example 5.39. Let $f : [l, u] \subseteq \mathbf{R} \to \mathbf{R}^2$ be a vector of functions satisfying all assumptions of Lemmas 5.36 and 5.37. Equations (5.8) and (5.9) can be solved analytically for vectors of functions f, where f_1 is of the form x^2, x^3 or $\sqrt{x}$, for instance. For $f_1 = x^2$ consider any function f_2 and $[l, u] \subseteq \mathbf{R}$ satisfying the assumptions of Lemmas 5.36 and 5.37. Then, the simultaneous convex hull $\mathcal{Q}_D[f]$ is the intersection of the constraints $(x, z_1) \in \mathcal{Q}_{[l,u]}[f_1] = \{(x, z) \mid f_1(x) \leq z_1 \leq \frac{f_1(u) - f_1(l)}{u - l}\,(x - l) + f_1(l)\}$,

$$z_2 \leq \frac{f_2(u)(z_1 - x^2) + f_2\left(\frac{ux - z_1}{u - x}\right)(u - x)^2}{z_1 + u^2 - 2ux} \quad \text{and} \quad z_2 \geq \frac{f_2(l)(z_1 - x^2) + f_2\left(\frac{z_1 - lx}{x - l}\right)(x - l)^2}{l^2 - 2lx + z_1}. \tag{5.10}$$

The constraints in Equation (5.10) correspond to all possible supporting hyperplanes on $Q_D[f]$ so that the constraint $(x, z_1) \in Q_{[l,u]}[f_1]$ is redundant and can be removed from the description of $Q_D[f]$. One can verify that the derived description is identical to the one for the moment curve if further $f_2 := x^3$ (cf. Subsection 5.1.1). ◇

Computational results of our proposed relaxations are presented after the discussion of the vector of three univariate convex functions.

5.3.2. Three Univariate Convex Functions

The previous analysis of a vector of two univariate functions showed the strength of the derived relaxations but already indicated some technical difficulties regarding the analytical derivation of the involved objects. In this subsection we extend these methods to a vector of three univariate convex functions in order to deduce improved relaxations but also to emphasize the technical limitations of our approach.

We consider a vector of three univariate convex functions $f : [l, u] \subseteq \mathbf{R} \to \mathbf{R}^3$ such that $\alpha^\top f$ possesses at most two inflection points over $[l, u]$ for all $\alpha \in \mathbf{R}^3$. Besides the convex, concave, convex-concave, and concave-convex cases, we further encounter functions whose convexity pattern is *convex-concave-convex*, i.e., they are first strictly convex, then strictly concave, and finally strictly convex again, and *concave-convex-concave*, i.e., they are first strictly concave, then strictly convex, and finally strictly concave.

Initially, we focus on C_{vex}. A given $\alpha \in \mathbf{R}^3$ belongs to C_{vex} if and only if $(\alpha^\top f)''(x) \geq 0$ for all $x \in [l, u]$. This expression is equivalent to $\alpha_1 \geq -\alpha_2 \frac{f_2''(x)}{f_1''(x)} - \alpha_3 \frac{f_3''(x)}{f_1''(x)}$ for all $x \in [l, u]$ if f_1 is strictly convex with $f_1''(x) > 0$ for all $x \in [l, u]$. To limit the number of inflection points of $\alpha^\top f$ to at most 2, we define $t[\alpha_2, \alpha_3](x) := -\alpha_2 \frac{f_2''(x)}{f_1''(x)} - \alpha_3 \frac{f_3''(x)}{f_1''(x)}$ and restrict $t[\alpha_2, \alpha_3](x)$ to be (i) strictly monotone increasing, (ii) strictly monotone decreasing, (iii) first strictly monotone increasing, then strictly monotone decreasing, or (iv) first strictly monotone decreasing, then strictly monotone increasing. The equation $\alpha_1 = t[\alpha_2, \alpha_3](x)$ possesses then at most two roots over $[l, u]$ each of which is equivalent to an inflection point of $\alpha^\top f$. Therefore, we analyze $(t[\alpha_2, \alpha_3])'(x) = -\alpha_2 \left(\frac{f_2''(x)}{f_1''(x)}\right)' - \alpha_3 \left(\frac{f_3''(x)}{f_1''(x)}\right)'$ and, in particular, the

following quotient of derivatives

$$L(x) := \frac{(f_3''(x)/f_1''(x))'}{(f_2''(x)/f_1''(x))'} = \frac{f_1''(x)f_3'''(x) - f_1'''(x)f_3''(x)}{f_1''(x)f_2'''(x) - f_1'''(x)f_2''(x)}.$$

If $L(x)$ is strictly monotone increasing, we show that $t[\alpha_2, \alpha_3](x)$ satisfies the required monotonicity properties and $(\alpha^\intercal f)''(x)$ exhibits at most two roots over $[l, u]$. Depending on these properties we can compute the minimal α_1 for a given pair (α_2, α_3) such that $\alpha^\intercal f$ is convex over $[l, u]$, i.e., the minimal α_1 with $\alpha_1 \geq t[\alpha_2, \alpha_3](x)$ for all $x \in [l, u]$.

Lemma 5.40. *Let $f : [l, u] \subseteq \mathbf{R} \to \mathbf{R}^3$ be a vector of four times continuously differentiable functions. Assume that (i) $f_i''(x) > 0$, $i = 1, 2, 3$, (ii) $(f_i''/f_1'')'(x) > 0$, $i = 2, 3$, and (iii) $L'(x) > 0$ for all $x \in [l, u]$. Define $a^\star = \frac{f_1''(l)f_3''(u) - f_1''(u)f_3''(l)}{f_1''(l)f_2''(u) - f_1''(u)f_2''(l)}$.*

(a) The function $t[\alpha_2, \alpha_3](x)$ exhibits four patterns of strict monotonicity over $[l, u]$:
- For $(\alpha_2, \alpha_3) \in \text{cone}(\{(L(l), -1), (-L(u), 1)\})$ increasing .
- For $(\alpha_2, \alpha_3) \in \text{int}(\text{cone}(\{(-L(u), 1), (-L(l), 1)\}))$ first increasing, then decreasing.
- For $(\alpha_2, \alpha_3) \in \text{cone}(\{(-L(l), 1), (L(u), -1)\})$ decreasing.
- For $(\alpha_2, \alpha_3) \in \text{int}(\text{cone}(\{(L(u), -1), (L(l), -1)\}))$ first decreasing, then increasing.

(b) If $(\alpha_2, \alpha_3) \in \text{cone}(\{(-L(u), 1), (a^\star, -1)\})$, $\max\{t[\alpha_2, \alpha_3](x) \mid x \in [l, u]\} = t[\alpha_2, \alpha_3](u)$.

(c) If $(\alpha_2, \alpha_3) \in \text{cone}(\{(-L(l), 1), (a^\star, -1)\})$, $\max\{t[\alpha_2, \alpha_3](x) \mid x \in [l, u]\} = t[\alpha_2, \alpha_3](l)$.

(d) If $(\alpha_2, 1) \in \text{int}(\text{cone}(\{(-L(u), 1), (-L(l), 1)\}))$, there is a unique $\bar{x} \in (l, u)$ with $\alpha_2 = -L(\bar{x})$ and $\max\{t[\alpha_2, \alpha_3](x) \mid x \in [l, u]\} = t[\alpha_2, \alpha_3](\bar{x})$.

(e) The space $\mathbf{R}^2$ can be represented as the union of $\text{cone}(\{(-L(u), 1), (a^\star, -1)\})$, $\text{cone}(\{(-L(l), 1), (a^\star, -1)\})$, and $\text{int}(\text{cone}(\{(-L(u), 1), (-L(l), 1)\}))$.

Proof. (a) The monotonicity patterns are implied by the monotonicity of $L(x)$ (condition (iii)). For instance, $(t[\alpha_2, \alpha_3])'(x) = -\alpha_2(f_2''(x)/f_1''(x))' - \alpha_3(f_3''(x)/f_1''(x))' \geq 0$ can be reformulated, using condition (ii), into $-\alpha_2 - \alpha_3 L(x) \geq 0$. If $\alpha_3 = -1$, then $(t[\alpha_2, \alpha_3])'(x) \geq 0$ if and only if $L(x) \geq \alpha_2$ for all $x \in [l, u]$. By monotonicity of $L(x)$, this is equivalent to $L(l) \geq \alpha_2$.

(b) We show that $\max\{t[\alpha_2,\alpha_3](x) \mid x \in [l,u]\} = t[\alpha_2,\alpha_3](u)$ for $(\alpha_2,\alpha_3) = (-L(u),1)$ and $(\alpha_2,\alpha_3) = (a^\star,-1)$ so that $\max\{t[\alpha_2,\alpha_3](x) \mid x \in [l,u]\} = t[\alpha_2,\alpha_3](u)$ for all $(\alpha_2,\alpha_3) \in \operatorname{cone}(\{(-L(u),1),(a^\star,-1\})$. In case of $(\alpha_2,\alpha_3) = (-L(u),1)$ this is implied by (a). For $(\alpha_2,\alpha_3) = (a^\star,-1)$ the function $t[\alpha_2,\alpha_3](x)$ can exhibit three possible monotonicity patterns as $\alpha_3 = -1$: Strictly monotone increasing, strictly monotone decreasing, or first strictly monotone decreasing, then strictly monotone increasing. The maximum is thus attained at $x = l$ or $x = u$. One can check that $t[a^\star,-1](l) = -\frac{f_2''(l)f_3''(u)-f_2''(u)f_3''(l)}{f_1''(l)f_2''(u)-f_1''(u)f_2''(l)} = t[a^\star,-1](u)$ which implies that $t[\alpha_2,\alpha_3](x)$ is first strictly decreasing, then strictly increasing. Thus, $\max\{t[\alpha_2,\alpha_3](x) \mid x \in [l,u]\} = t[\alpha_2,\alpha_3](l) = t[\alpha_2,\alpha_3](u)$.

(c) We can apply the same arguments as for (b).

(d) If $(\alpha_2,1) \in \operatorname{int}(\operatorname{cone}(\{(-L(u),1),(-L(l),1)\}))$, the strict monotonicity of $L(x)$ implies that there is a unique $\bar{x} \in (l,u)$ with $\alpha_2 = -L(\bar{x})$. Moreover, $t[\alpha_2,\alpha_3](x)$ is first strictly increasing, then strictly decreasing. Thus, there is a unique maximizer $x^* \in (l,u)$ satisfying the first order optimality condition $(t[\alpha_2,\alpha_3])'(x^*) = -\alpha_2(f_2''(x^*)/f_1''(x^*))' - 1(f_3''(x^*)/f_1''(x^*))' = 0$ which is equivalent to $-\alpha_2 - L(x^*) = 0$. This condition holds if and only if $L(x^*) = -\alpha_2 = L(\bar{x})$ and thus, if and only if $x^* = \bar{x}$.

(e) The conic combination of three vectors $v^1,v^2,v^3 \in \mathbf{R}^2$ spans $\mathbf{R}^2$ if $-v^1$ is in the interior of $\operatorname{cone}(\{v^2,v^3\})$. We showed in (b) that $t[a^\star,-1]$ is first strictly decreasing, then strictly increasing. Thus, $-t[a^\star,-1] = t[-a^\star,1]$ is first strictly increasing, then strictly decreasing so that (a) implies $-(a^\star,-1) \in \operatorname{int}(\operatorname{cone}(\{(-L(u),1),(-L(l),1)\}))$. □

The previous analysis yields C_{vex}.

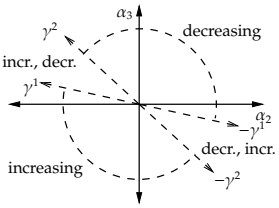
Theorem 5.41. *Let $f : [l,u] \subseteq \mathbf{R} \to \mathbf{R}^3$ be a vector of four times continuously differentiable functions. Assume that (i) $f_i''(x) > 0$, $i = 1,2,3$, (ii) $(f_i''/f_1'')'(x) > 0$, $i = 2,3$, and (iii) $L'(x) > 0$ for all $x \in [l,u]$. Then, $C_{\text{vex}} = \operatorname{cone}(\{\alpha^1_{\text{vex}},\alpha^2_{\text{vex}},\alpha^3_{\text{vex}}\} \cup A_{\text{vex}})$, where*

$$\begin{aligned}
\alpha^1_{\text{vex}} &:= (t[-L(u),1](u)\ ,\ -L(u)\ ,\ 1),\\
\alpha^2_{\text{vex}} &:= (t[-L(l),1](l)\ ,\ -L(l)\ ,\ 1),\\
\alpha^3_{\text{vex}} &:= (t[\tfrac{f_1''(l)f_3''(u)-f_1''(u)f_3''(l)}{f_1''(l)f_2''(u)-f_1''(u)f_2''(l)},-1](u)\ ,\ \tfrac{f_1''(l)f_3''(u)-f_1''(u)f_3''(l)}{f_1''(l)f_2''(u)-f_1''(u)f_2''(l)}\ ,\ -1),\\
A_{\text{vex}} &:= \{(t[-L(x),1](x)\ ,\ -L(x)\ ,\ 1) \in \mathbf{R}^3 \mid x \in (l,u)\}.
\end{aligned}$$

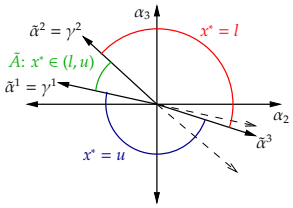
Proof. A given $\alpha \in \mathbf{R}^3$ belongs to C_{vex} if and only if $(\alpha^\intercal f)''(x) \geq 0$ for all $x \in [l,u]$. Due to assumption (i), this expression is equivalent to $\alpha_1 \geq \max\{t[\alpha_2,\alpha_3](x) \mid x \in [l,u]\}$. We can infer from Lemma 5.40 (e) that (α_2,α_3) is contained in one of the cones defined in Lemma 5.40 (b)-(d), where we also determine $\max\{t[\alpha_2,\alpha_3](x) \mid x \in [l,u]\}$. □

Before we proceed with C_{poly} we give an example for C_{vex}.

Example 5.42. Consider $f = (x^3, x^5, x^6)$ restricted to $[l,u] := [1/2, 2]$ which satisfies all requirements of Theorem 5.41. According to Lemma 5.40 the monotonicity behavior of the auxiliary function $t[\alpha_2,\alpha_3](x)$ leads to a subdivision of the $(\alpha_2,\alpha_3) \in \mathbf{R}^2$ space into 4 subdomains which are indicated in Figure 5.7 (a), where the vectors γ^1 and γ^2 are positive multiples of $(-L(u),1) = (-9/2, 1)$ and $(-L(l),1) = (-9/8, 1)$, respectively.



(a) Monotonicity patterns of $t[\alpha_2,\alpha_3](x)$.



(b) Maximizer x^* of $t[\alpha_2,\alpha_3](x)$.

Figure 5.7.: Figure (a) depicts the subdivision regarding the monotonicity of $t[\alpha_2,\alpha_3](x)$ over $[l,u]$. This leads to a subdivision regarding the maximizer of $\max\{t[\alpha_2,\alpha_3](x) \mid x \in [l,u]\}$ in Figure (b) and yields C_{vex}.

Given the monotonicity we can solve $\max\{t[\alpha_2,\alpha_3](x) \mid x \in [l,u]\}$ whose optimal objective function value equals the minimal α_1 for given (α_2,α_3) such that $\alpha^\intercal f$ is convex over $[l,u]$. We obtain $C_{\text{vex}} = \text{cone}(\{\alpha^1_{\text{vex}}, \alpha^2_{\text{vex}}, \alpha^3_{\text{vex}}\} \cup A_{\text{vex}})$ with

$$\alpha^1_{\text{vex}} = (20\,,\, -9/2\,,\, 1), \quad \alpha^2_{\text{vex}} = (5/16\,,\, -9/8\,,\, 1)\,, \quad \alpha^3_{\text{vex}} = (-2\,,\, 63/20\,,\, -1)\,,$$

and $A_{\text{vex}} = \{(5/2\, x^3\,,\, -9/4\, x\,,\, 1) \mid x \in (l,u)\}$. The projection of the vectors α^i_{vex} and the set A_{vex} onto the (α_2,α_3)-space is denoted by $\tilde{\alpha}^i$ and $\tilde{A}$, respectively, and illustrated in Figure 5.7 (b). The figure shows that $\tilde{A}$ corresponds to the (α_2,α_3)-area over which $t[\alpha_2,\alpha_3](x)$ is first strictly monotone increasing

and then decreasing. Moreover, if $t[\alpha_2, \alpha_3](x)$ is first strictly monotone decreasing and then increasing, $\tilde{\alpha}^3$ clearly separates the area, where the maximum of $t[\alpha_2, \alpha_3](x)$ is attained at $x = l$ and $x = u$. ◇

The monotonicity patterns of the auxiliary function $t[\alpha_2, \alpha_3](x)$ lead to convexity patterns for $\alpha^\intercal f$ depending on α_1, which are depicted in Figure 5.8, where the subdivision of the space corresponds to Figure 5.7 (a). In order to determine C_{poly}, the figure indicates that we need suitable criteria for convex-concave-convex and concave-convex-concave functions such that their convex envelopes are vertex polyhedral. This is the subject of the remainder of this section.

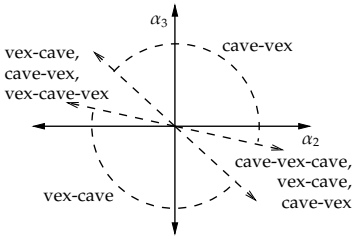


Figure 5.8.: Possible convexity patterns besides strictly convex (vex) and strictly concave (cave).

The unique candidate for a vertex polyhedral convex envelope of a univariate, continuously differentiable function $g : [l, u] \subseteq \mathbf{R} \to \mathbf{R}$ is the following affine function

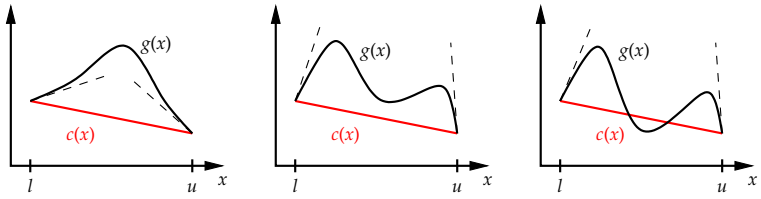
$$c(x) := \tfrac{g(u)-g(l)}{u-l}(x - l) + g(l).$$

Thus, a necessary condition for having a vertex polyhedral convex envelope is

$$g'(l) \quad \geq \quad \tfrac{g(u)-g(l)}{u-l} \quad \geq \quad g'(u), \tag{5.11}$$

which is indicated in Figure 5.9. The dashed lines correspond to the slopes $g'(l)$ and $g'(u)$, respectively, and the red line represents the slope $\frac{g(u)-g(l)}{u-l}$ of $c(x)$. If g is concave-convex or convex-concave, Condition (5.11) is also sufficient for having a vertex polyhedral convex-envelope (cf. Observation 5.30). This is also true for convex-concave-convex functions as

indicated in Figure 5.9 (a).



(a) Vex-Cave-Vex. (b) Cave-Vex-Cave. (c) Cave-Vex-Cave.

Figure 5.9.: The dashed lines indicate the slopes $g'(l)$ and $g'(u)$ while the red line represents the slope of the potential convex envelope.

Observation 5.43. *Let $g : [l,u] \subseteq \mathbf{R} \to \mathbf{R}$ be a two times continuously differentiable function which is convex-concave-convex. Then, $\mathrm{vex}_{[l,u]}[g]$ is vertex polyhedral if and only if Condition* (5.11) *is satisfied.*

In contrast to this, Condition (5.11) is not sufficient for concave-convex-concave functions to have a vertex polyhedral convex-envelope as illustrated in Figure 5.9 (c). To guarantee vertex polyhedrality, we analyze the minimal difference between $g(x)$ and its potential vertex polyhedral convex envelope $c(x)$. If this minimum is nonnegative, $c(x)$ is the convex envelope of g.

Observation 5.44. *Let $g : [l,u] \subseteq \mathbf{R} \to \mathbf{R}$ be a two times continuously differentiable function which is concave-convex-concave. Then, $\mathrm{vex}_{[l,u]}[g] = c(x)$ if and only if there is an $\bar{x} \in (l,u)$ with (i) $g'(\bar{x}) = \frac{g(u)-g(l)}{u-l}$ and (ii) $g''(\bar{x}) > 0$ satisfying $g(\bar{x}) \geq c(\bar{x})$. If such an $\bar{x}$ exists, it is unique.*

We interpret Observations 5.43 and 5.44 for $g = \alpha^\intercal f$ with $\alpha \in \mathbf{R}^3$ and $f : [l,u] \subseteq \mathbf{R} \to \mathbf{R}^3$ in order to derive conditions on α such that $\alpha^\intercal f$ possesses a vertex polyhedral convex envelope. Condition (5.11) reads $(\alpha^\intercal f)'(l) - \frac{\alpha^\intercal f(u)-\alpha^\intercal f(l)}{u-l} \geq 0$ and $\frac{\alpha^\intercal f(u)-\alpha^\intercal f(l)}{u-l} - (\alpha^\intercal f)'(u) \geq 0$, which is equivalent to

$$\sum_{i=1}^{3} -\alpha_i \left(f_i(u) - f_i(l) - (u-l) f_i'(l) \right) \geq 0 \quad \text{and} \quad \sum_{i=1}^{3} \alpha_i \left(f_i(l) - f_i(u) - (l-u) f_i'(u) \right) \leq 0,$$

and thus, equivalent to

$$\alpha_1 \leq -\alpha_2 T_u^2(l) - \alpha_3 T_u^3(l) \quad \text{and} \quad \alpha_1 \leq -\alpha_2 T_l^2(u) - \alpha_3 T_l^3(u), \tag{5.12}$$

where

$$T_l^i(x) := \begin{cases} f_i''(l)/f_1''(l), & x = l, \\ \frac{f_i(l)-f_i(x)-(l-x)f_i'(x)}{f_1(l)-f_1(x)-(l-x)f_1'(x)}, & x > l, \end{cases} \quad T_u^i(x) := \begin{cases} \frac{f_i(u)-f_i(x)-(u-x)f_i'(x)}{f_1(u)-f_1(x)-(u-x)f_1'(x)}, & x < u, \\ f_i''(u)/f_1''(u), & x = u, \end{cases}$$

for $i = 2, 3$. Note that the functions $T_l^i(x)$ and $T_u^i(x)$ are defined analogously to $T_l(x)$ and $T_u(x)$ in the previous section.

Observation 5.44 implies for $g(x) = \alpha^\intercal f(x)$ that we have to guarantee nonnegativity of the minimal difference between $\alpha^\intercal f$ and its potential convex envelope $c(x) = \frac{\alpha^\intercal f(u)-\alpha^\intercal f(l)}{u-l}(x-l) + \alpha^\intercal f(l)$, i.e., for all $x \in [l, u]$ it has to hold that

$$\sum_{i=1}^{3} \alpha_i \left(\tfrac{f_i(u)-f_i(l)}{u-l}(x-l) + f_i(l) - f_i(x) \right) \leq 0. \tag{5.13}$$

The expression in the big parenthesis represents the difference between the secant of a strictly convex function f_i and the function itself. At $x \in \{l, u\}$ the secant and the function coincide so that the difference is zero and Equation (5.13) is satisfied. If $x \in (l, u)$, the difference is strictly positive and we can reformulate Equation (5.13) as

$$\alpha_1 + \sum_{i=2}^{3} \alpha_i \underbrace{\frac{\frac{f_i(u)-f_i(l)}{u-l}(x-l) + f_i(l) - f_i(x)}{\frac{f_1(u)-f_1(l)}{u-l}(x-l) + f_1(l) - f_1(x)}}_{=:S^i(x)} \leq 0. \tag{5.14}$$

Thus, we derive the condition $\alpha_1 \leq \inf\{s[\alpha_2, \alpha_3](x) \mid x \in (l, u)\}$, where $s[\alpha_2, \alpha_3](x) := -\alpha_2 S^2(x) - \alpha_3 S^3(x)$. We define $S^i(l) := \lim_{x \to l} S^i(x) = T_u^i(l)$ and $S^i(u) := \lim_{x \to u} S^i(x) = T_l^i(u)$. If the minimizer of $\min\{s[\alpha_2, \alpha_3](x) \mid x \in [l, u]\}$ is attained at $x = l$ or $x = u$, we obtain

$$\alpha_1 \leq -\alpha_2 T_u^2(l) - \alpha_3 T_u^3(l) \quad \text{or} \quad \alpha_1 \leq -\alpha_2 T_l^2(u) - \alpha_3 T_l^3(u),$$

respectively. These constraints are already given by the necessary Condition (5.12).

Summarizing the derived conditions, we conclude that $\alpha \in C_{\text{poly}}$ if and

only if

$$\alpha_1 \leq \min \left\{ \begin{array}{l} \min\{s[\alpha_2, \alpha_3](x) \mid x \in [l,u]\}, \\ -\alpha_2 T_u^2(l) - \alpha_3 T_u^3(l), \\ -\alpha_2 T_l^2(u) - \alpha_3 T_l^3(u) \end{array} \right\}. \tag{5.15}$$

In particular, α is in the boundary of C_{poly} if and only if equality holds in Equation (5.15). The solution of the optimization problem in Equation (5.15) leads to the description of C_{poly}.

Theorem 5.45. *Let $f : [l,u] \subseteq \mathbf{R} \to \mathbf{R}^3$ be a vector of four times continuously differentiable functions. Assume that (i) $f_i''(x) > 0$, $i = 1,2,3$, (ii) $(f_i''/f_1'')'(x) > 0$, $i = 2,3$, and (iii) $L'(x) > 0$ for all $x \in [l,u]$. Then, $C_{\text{poly}} = \text{cone}(\{\beta^1_{\text{poly}}, \beta^2_{\text{poly}}, \beta^3_{\text{poly}}\} \cup B_{\text{poly}})$, where*

$$\begin{aligned}
\beta^1_{\text{poly}} &:= \left(-\tfrac{T_l^3(u) - f_3''(u)/f_1''(u)}{T_l^2(u) - f_2''(u)/f_1''(u)} T_l^2(u) + T_l^3(u) \,,\; \tfrac{T_l^3(u) - f_3''(u)/f_1''(u)}{T_l^2(u) - f_2''(u)/f_1''(u)} \,,\; -1 \right), \\
\beta^2_{\text{poly}} &:= \left(-\tfrac{T_u^3(l) - f_3''(l)/f_1''(l)}{T_u^2(l) - f_2''(l)/f_1''(l)} T_u^2(l) + T_u^3(l) \,,\; \tfrac{T_u^3(l) - f_3''(l)/f_1''(l)}{T_u^2(l) - f_2''(l)/f_1''(l)} \,,\; -1 \right), \\
\beta^3_{\text{poly}} &:= \left(\tfrac{T_l^3(u) - T_u^3(l)}{T_l^2(u) - T_u^2(l)} T_u^2(l) - T_u^3(l) \,,\; -\tfrac{T_l^3(u) - T_u^3(l)}{T_l^2(u) - T_u^2(l)} \,,\; 1 \right), \\
B_{\text{poly}} &:= \left\{ \left(-\tfrac{(S^3)'(x)}{(S^2)'(x)} S^2(x) + S^3(x) \,,\; \tfrac{(S^3)'(x)}{(S^2)'(x)} \,,\; -1 \right) \mid x \in (l,u) \right\}.
\end{aligned}$$

We analyze the subproblem $\min\{s[\alpha_2, \alpha_3](x) \mid x \in [l,u]\}$ of the optimization problem in Equation (5.15) before we prove the theorem. For this, we investigate the functions $S^i(x)$ in order to determine the monotonicity patterns of $s[\alpha_2, \alpha_3](x) = -\alpha_2 S^2(x) - \alpha_3 S^3(x)$.

Lemma 5.46. *Let $f : [l,u] \subseteq \mathbf{R} \to \mathbf{R}^3$ be a vector of four times continuously differentiable functions. Assume that (i) $f_i''(x) > 0$, $i = 1,2,3$, (ii) $(f_i''/f_1'')'(x) > 0$, $i = 2,3$, and (iii) $L'(x) > 0$ for all $x \in (l,u)$. Then, $(S^i)'(x) > 0$, $i = 2,3$, and $((S^3)'/(S^2)')'(x) > 0$ for all $x \in (l,u)$. Moreover,*

$$\lim_{x \to l} \tfrac{(S^3)'(x)}{(S^2)'(x)} = \tfrac{T_u^3(l) - f_3''(l)/f_1''(l)}{T_u^2(l) - f_2''(l)/f_1''(l)} = (\beta^2_{\text{poly}})_2 \,,\; \lim_{x \to u} \tfrac{(S^3)'(x)}{(S^2)'(x)} = \tfrac{T_l^3(u) - f_3''(u)/f_1''(u)}{T_l^2(u) - f_2''(u)/f_1''(u)} = (\beta^1_{\text{poly}})_2.$$

Proof. To simplify notation, we introduce $a_i(x) := \frac{f_i(u) - f_i(l)}{u - l}(x - l) + f_i(l) - f_i(x)$, i.e., $S^i(x) = a_i(x)/a_1(x)$ and $(S^i)'(x) = \frac{a_1(x)a_i'(x) - a_1'(x)a_i(x)}{(a_1(x))^2} = \frac{a_i'(x) - a_1'(x)S^i(x)}{a_1(x)}$, $i = 2,3$.

We show that there is no $\bar{x} \in (l,u)$ with $(S^2)'(\bar{x}) = 0$. Then, Lemma 5.34 and the definition of $S^i(x)$ imply that $S^i(l) = T^i_u(l) < T^i_l(u) = S^i(u)$ so that $S^i(x)$ can only be strictly monotone increasing. Assume that there exists an $\bar{x} \in (l,u)$ with $(S^2)'(\bar{x}) = 0$ and define $\alpha := (-\alpha_2 S^2(\bar{x}), \alpha_2, 0)$ with $\alpha_2 \in \mathbf{R} \setminus \{0\}$. One can check that

$$\alpha^\intercal f(x) = c(x) \quad \text{for } x \in \{l, \bar{x}, u\} \quad \text{and} \quad (\alpha^\intercal f)'(\bar{x}) = c'(\bar{x}). \tag{5.16}$$

In Lemmas 5.30 and 5.31 we proved that $\alpha^\intercal f(x)$ with $\alpha_3 = 0$ is either strictly convex, strictly concave, convex-concave or concave-convex. In none of these cases $\alpha^\intercal f(x)$ has a shape which allows for an affine function $c(x)$ with the properties in Equation (5.16) and hence, there is no $\bar{x} \in (l,u)$ that satisfies the assumptions. Analogously, $(S^3)'(x) > 0$ for all $x \in (l,u)$ can be proven.

To prove that $\left(\frac{(S^3)'(\bar{x})}{(S^2)'(\bar{x})}\right)' = \left(\frac{a_3'(\bar{x}) - a_1'(\bar{x}) S^3(\bar{x})}{a_2'(\bar{x}) - a_1'(\bar{x}) S^2(\bar{x})}\right)'$ is positive for all $\bar{x} \in (l,u)$, we reformulate the expression as

$$\begin{aligned}
&\frac{\left(a_3''(\bar{x}) - a_1''(\bar{x})\, S^3(\bar{x}) - a_1'(\bar{x})\, (S^3)'(\bar{x})\right)\left(a_2'(\bar{x}) - a_1'(\bar{x})\, S^2(\bar{x})\right) - \left(a_3'(\bar{x}) - a_1'(\bar{x})\, S^3(\bar{x})\right)\left(a_2''(\bar{x}) - a_1''(\bar{x})\, S^2(\bar{x}) - a_1'(\bar{x})\, (S^2)'(\bar{x})\right)}{(a_2'(\bar{x}) - a_1'(\bar{x})\, S^2(\bar{x}))^2} \\
&= \frac{\left(a_3''(\bar{x}) - a_1''(\bar{x})\, S^3(\bar{x}) - a_1'(\bar{x})\, (S^3)'(\bar{x})\right) - \frac{(S^3)'(\bar{x})}{(S^2)'(\bar{x})}\left(a_2''(\bar{x}) - a_1''(\bar{x})\, S^2(\bar{x}) - a_1'(\bar{x})\, (S^2)'(\bar{x})\right)}{a_2'(\bar{x}) - a_1'(\bar{x})\, S^2(\bar{x})} \\
&= \frac{\left(-\frac{(S^3)'(\bar{x})}{(S^2)'(\bar{x})}\, S^2(\bar{x}) + S^3(\bar{x})\right)(-a_1''(\bar{x})) + \frac{(S^3)'(\bar{x})}{(S^2)'(\bar{x})}\,(-a_2''(\bar{x})) - (-a_3''(\bar{x}))}{a_2'(\bar{x}) - a_1'(\bar{x})\, S^2(\bar{x})} \\
&= \frac{(\alpha^\intercal f)''(\bar{x})}{a_2'(\bar{x}) - a_1'(\bar{x})\, S^2(\bar{x})},
\end{aligned} \tag{5.17}$$

where the last equation follows from $a_i''(x) = -f_i''(x)$ and

$$\alpha := \left(-\frac{(S^3)'(\bar{x})}{(S^2)'(\bar{x})} S^2(\bar{x}) + S^3(\bar{x}),\ \frac{(S^3)'(\bar{x})}{(S^2)'(\bar{x})},\ -1\right).$$

A positive sign of the denominator $a_2'(\bar{x}) - a_1'(\bar{x})\, S^2(\bar{x})$ in Equation (5.17) is implied by $(S^i)'(\bar{x}) = \frac{a_i'(\bar{x}) - a_1'(\bar{x}) S^i(\bar{x})}{a_1(\bar{x})} > 0$, which we proved before, and $a_1(\bar{x}) > 0$. It remains to show that $(\alpha^\intercal f)''(\bar{x}) > 0$. For this, we investigate the function $\alpha^\intercal f$. It can be shown that

$$\alpha^\intercal f(x) = c(x) \quad \text{for } x \in \{l, \bar{x}, u\} \quad \text{and} \quad (\alpha^\intercal f)'(\bar{x}) = c'(\bar{x}). \tag{5.18}$$

These conditions can only be met if $\alpha^\intercal f(x)$ is either convex-concave-

convex or concave-convex-concave. As $\alpha_3 = -1$, $\alpha^\intercal f$ has to be concave-convex-concave (cf. Figure 5.8) and $(\alpha^\intercal f)''(\bar{x}) > 0$. Thus, ${}^{(S^3)'(x)}/_{(S^2)'(x)}$ is strictly monotone increasing.

The limits of ${}^{(S^3)'(x)}/_{(S^2)'(x)}$ as x approaches l or u follow from L'Hôpital's rule and are $(\beta^2_{\text{poly}})_2$ and $(\beta^1_{\text{poly}})_2$, respectively. □

In Lemma 5.40 we computed the maximizer of the auxiliary function $t[\alpha_2, \alpha_3](x) = -\alpha_2 {}^{f_2''(x)}/_{f_1''(x)} - \alpha_3 {}^{f_3''(x)}/_{f_1''(x)}$ over $[l, u]$ using the fact that $L(x) = \frac{(f_3''(x)/f_1''(x))'}{(f_2''(x)/f_1''(x))'}$ is strictly monotone increasing. Analogously, we can now determine the minimizer of $s[\alpha_2, \alpha_3](x) = -\alpha_2 S^2(x) - \alpha_3 S^3(x)$ since ${}^{(S^3)'(x)}/_{(S^2)'(x)}$ is strictly monotone increasing as shown in the previous lemma.

Lemma 5.47. *Let $f : [l, u] \subseteq \mathbf{R} \to \mathbf{R}^3$ be a vector of four times continuously differentiable functions. Assume that (i) $f_i''(x) > 0$, $i = 1, 2, 3$, (ii) $(f_i''/f_1'')'(x) > 0$, $i = 2, 3$, and (iii) $L'(x) > 0$ for all $x \in [l, u]$. Let $\tilde{\beta}^i_{\text{poly}} := ((\beta^i_{\text{poly}})_2, (\beta^i_{\text{poly}})_3)$, $i = 1, 2, 3$.*

(a) The function $s[\alpha_2, \alpha_3](x)$ exhibits four patterns of strict monotonicity over $[l, u]$:
- Increasing for $(\alpha_2, \alpha_3) \in \text{cone}(\{\tilde{\beta}^2_{\text{poly}}, -\tilde{\beta}^1_{\text{poly}}\})$.
- First increasing, then decreasing for $(\alpha_2, \alpha_3) \in \text{int}(\text{cone}(\{-\tilde{\beta}^1_{\text{poly}}, -\tilde{\beta}^2_{\text{poly}}\}))$.
- Decreasing for $(\alpha_2, \alpha_3) \in \text{cone}(\{-\tilde{\beta}^2_{\text{poly}}, \tilde{\beta}^1_{\text{poly}}\})$.
- First decreasing, then increasing for $(\alpha_2, \alpha_3) \in \text{int}(\text{cone}(\{\tilde{\beta}^1_{\text{poly}}, \tilde{\beta}^2_{\text{poly}}\}))$.

(b) If $(\alpha_2, \alpha_3) \in \text{cone}(\{\tilde{\beta}^2_{\text{poly}}, \tilde{\beta}^3_{\text{poly}}\})$, then $\min\{s[\alpha_2, \alpha_3](x) \mid x \in [l, u]\} = s[\alpha_2, \alpha_3](l)$.

(c) If $(\alpha_2, \alpha_3) \in \text{cone}(\{\tilde{\beta}^3_{\text{poly}}, \tilde{\beta}^1_{\text{poly}}\})$, then $\min\{s[\alpha_2, \alpha_3](x) \mid x \in [l, u]\} = s[\alpha_2, \alpha_3](u)$.

(d) If $(\alpha_2, -1) \in \text{int}(\text{cone}(\{\tilde{\beta}^1_{\text{poly}}, \tilde{\beta}^2_{\text{poly}}\}))$, there is a unique $\bar{x} \in (l, u)$ with $\alpha_2 = \frac{(S^3)'(\bar{x})}{(S^2)'(\bar{x})}$ and $\min\{s[\alpha_2, \alpha_3](x) \mid x \in [l, u]\} = s[\alpha_2, \alpha_3](\bar{x})$.

(e) The space $\mathbf{R}^2$ can be represented as the union of the cones $\text{cone}(\{\tilde{\beta}^2_{\text{poly}}, \tilde{\beta}^3_{\text{poly}}\})$, $\text{cone}(\{\tilde{\beta}^3_{\text{poly}}, \tilde{\beta}^1_{\text{poly}}\})$, and $\text{int}(\text{cone}(\{\{\tilde{\beta}^1_{\text{poly}}, \tilde{\beta}^2_{\text{poly}}\}))$.

Finally, we prove Theorem 5.45.

Proof of Theorem 5.45. A vector $\alpha \in \mathbf{R}^3$ belongs to the boundary of C_{poly} if and only if it satisfies Equation (5.15) with equality, i.e., at least one of the following constraints is active:

$$\alpha_1 \leq \min\{-\alpha_2 S^2(x) - \alpha_3 S^3(x) \mid x \in [l,u]\}, \tag{5.19}$$

$$\alpha_1 \leq -\alpha_2 T_u^2(l) - \alpha_3 T_u^3(l), \tag{5.20}$$

$$\alpha_1 \leq -\alpha_2 T_l^2(u) - \alpha_3 T_l^3(u). \tag{5.21}$$

Let $\tilde{\alpha}$ denote the projection of a vector $\alpha \in \mathbf{R}^3$ onto its 2nd and 3rd component, i.e., $\tilde{\alpha} = (\alpha_2, \alpha_3)$. To identify the boundary of C_{poly}, we determine the minimal α_1 for a given $\tilde{\alpha} \in \mathbf{R}^2$ such that $\alpha^\intercal f$ exhibits a vertex polyhedral convex envelope. Lemma 5.47 (e) implies that a given $\tilde{\alpha} \in \mathbf{R}^2$ belongs to one of the three cones defined in Lemma 5.47 (b)-(d). The same lemma also allows to solve subproblem $\min\{-\alpha_2 S^2(x) - \alpha_3 S^3(x) \mid x \in [l,u]\}$ (cf. Equation (5.19)). Subsequently, we determine the minimal α_1 over each cone which yields the boundary of C_{poly}.

If $\tilde{\alpha} \in \text{cone}(\{\tilde{\beta}^2_{\text{poly}}, \tilde{\beta}^3_{\text{poly}}\})$ or $\tilde{\alpha} \in \text{cone}(\{\tilde{\beta}^3_{\text{poly}}, \tilde{\beta}^1_{\text{poly}}\})$, the solution of the problem $\min\{s[\alpha_2, \alpha_3](x) \mid x \in [l,u]\}$ is attained at $x \in \{l,u\}$. Then, Equation (5.19) changes to $\alpha_1 \leq -\alpha_2 T_u^2(l) - \alpha_3 T_u^3(l)$ or $\alpha_1 \leq -\alpha_2 T_l^2(u) - \alpha_3 T_l^3(u)$, respectively, so that Equation (5.19) is redundant. Using Lemma 5.47 one can check that the β^i_{poly}, $i = 1,2,3$, satisfy the remaining conditions for vertex polyhedrality in Equations (5.20) and (5.21) in the following sense:

$$(\beta^1_{\text{poly}})_1 < -(\beta^1_{\text{poly}})_2\, T_u^2(l) - (\beta^1_{\text{poly}})_3\, T_u^3(l), \quad (\beta^1_{\text{poly}})_1 = -(\beta^1_{\text{poly}})_2\, T_l^2(u) - (\beta^1_{\text{poly}})_3\, T_l^3(u),$$
$$(\beta^2_{\text{poly}})_1 = -(\beta^2_{\text{poly}})_2\, T_u^2(l) - (\beta^2_{\text{poly}})_3\, T_u^3(l), \quad (\beta^2_{\text{poly}})_1 < -(\beta^2_{\text{poly}})_2\, T_l^2(u) - (\beta^2_{\text{poly}})_3\, T_l^3(u),$$
$$(\beta^3_{\text{poly}})_1 = -(\beta^3_{\text{poly}})_2\, T_u^2(l) - (\beta^3_{\text{poly}})_3\, T_u^3(l), \quad (\beta^3_{\text{poly}})_1 = -(\beta^3_{\text{poly}})_2\, T_l^2(u) - (\beta^3_{\text{poly}})_3\, T_l^3(u).$$

For each β^i_{poly}, $i = 1,2,3$, at least one of the constraints is active and it follows that the points β^i_{poly} are contained in the boundary of C_{poly}. This is also true for any conic combination of either β^1_{poly} and β^3_{poly}, or β^2_{poly} and β^3_{poly}. Thus, the points β^i_{poly}, $i = 1,2,3$, are the extreme rays of C_{poly} corresponding to the first two cones. It remains to determine the extreme rays corresponding to the last cone $\text{int}(\text{cone}(\{\{\tilde{\beta}^1_{\text{poly}}, \tilde{\beta}^2_{\text{poly}}\})))$.

Let $\tilde{\alpha} \in \text{int}(\text{cone}(\{\{\tilde{\beta}^1_{\text{poly}}, \tilde{\beta}^2_{\text{poly}}\})))$, where we can scale each $\tilde{\alpha}$ such that $\alpha_3 = -1$. According to Lemma 5.47 (d), there is an $\bar{x} \in (l,u)$ with $\alpha_2 =$

$(S^3)'(\bar{x})/(S^2)'(\bar{x})$. Therefore, consider

$$\alpha = \left(-\frac{(S^3)'(\bar{x})}{(S^2)'(\bar{x})} S^2(\bar{x}) + S^3(\bar{x}) \,,\, \frac{(S^3)'(\bar{x})}{(S^2)'(\bar{x})} \,,\, -1 \right)$$

and the resulting function $\alpha^\top f$. In the proof of Lemma 5.46 we showed that $\alpha^\top f$ is concave-convex-concave and that

$$\alpha^\top f(\bar{x}) = c(\bar{x}), \quad (\alpha^\top f)'(\bar{x}) = c'(\bar{x}), \quad (\alpha^\top f)''(\bar{x}) > 0.$$

Thus, Lemma 5.44 implies that the convex envelope of $\alpha^\top f$ is vertex polyhedral. Equations (5.20) and (5.21) represent the necessary conditions for vertex polyhedrality and must therefore be satisfied by α. Equations (5.19)–(5.21) reduce thus to Equation (5.19) which is satisfied with equality for all $\alpha \in B_{\text{poly}}$.

To conclude, we determined for all $\tilde{\alpha} \in \mathbf{R}^2$ the minimal α_1 such that $\text{vex}_D[\alpha^\top f]$ is vertex polyhedral, i.e., the boundary of C_{poly}. □

In Table 5.3 we give an overview of the characteristics of the extreme rays of C_{poly} given in Theorem 5.45 and revealed in Lemmas 5.46 and 5.47.

	Convexity of $g = \alpha^\top f$	$g'(l) - \frac{g(u)-g(l)}{u-l}$	$\frac{g(u)-g(l)}{u-l} - g'(u)$
$\alpha = \beta^1_{\text{poly}}$	concave-convex	> 0	$= 0$
$\alpha = \beta^2_{\text{poly}}$	convex-concave	$= 0$	> 0
$\alpha = \beta^3_{\text{poly}}$	convex-concave-convex	$= 0$	$= 0$
$\alpha \in B_{\text{poly}}$	concave-convex-concave	> 0	> 0

Table 5.3.: Characteristics of the extreme rays of C_{poly}.

The explicit descriptions of the cones C_{vex} and C_{poly} are now applied to construct basic relaxations of $Q_D[f]$ in case of a vector of three functions. Analogously to the vector of two functions, we propose a basic relaxation corresponding to the vectors α^i_{vex} and β^i_{vex} (defined in Theorems 5.41 and 5.45) which can be used as a strong initial relaxation in a branch-and-bound algorithm before further cuts are added. The strength of the basic relaxation is illustrated in the next example.

Example 5.48 (Example 5.42 continued). Let $f = (x^3, x^5, x^6)$ be restricted to $[l, u] := [1/2, 2]$. In Example 5.42 we computed $C_{\text{vex}} = \text{cone}(\{\alpha^1_{\text{vex}}, \alpha^2_{\text{vex}}, \alpha^3_{\text{vex}}\} \cup$

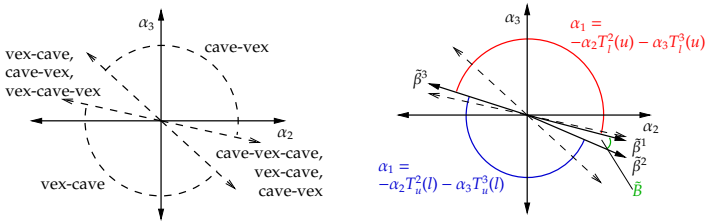
A_{vex}). Theorem 5.45 leads to $C_{\text{poly}} = \text{cone}(\{\beta^1_{\text{poly}}, \beta^2_{\text{poly}}, \beta^3_{\text{poly}}\} \cup B_{\text{poly}})$ with

$$\beta^1_{\text{poly}} = \left(-\tfrac{1900}{167}, \tfrac{1287}{334}, -1\right), \quad \beta^2_{\text{poly}} = \left(-\tfrac{475}{368}, \tfrac{423}{184}, -1\right), \quad \beta^3_{\text{poly}} = \left(\tfrac{421}{80}, -\tfrac{63}{20}, 1\right),$$

and

$$B_{\text{poly}} = \left\{\left(-\tfrac{4x^6+20x^5+67x^4+190x^3+67x^2+20x+4}{2(4x^3+20x^2+25x+5)}, \quad \tfrac{3(4x^4+20x^3+32x^2+35x+7)}{2(4x^3+20x^2+25x+5)}, \quad -1\right) \mid x \in (l,u)\right\}.$$

Besides being strictly convex or strictly concave, $\alpha^\intercal f$ can have the additional convexity patterns shown in Figure 5.10 (a). For a given (α_2, α_3) the patterns depend on the choice of α_1. In Figure 5.10 (b) we indicate the



(a) Possible convexity patterns.

(b) Subdivision of projected C_{poly}.

Figure 5.10.: Subdivision of $\mathbf{R}^2$ w.r.t. the possible convexity patterns of $\alpha^\intercal f$ besides strictly convex (vex) and strictly concave (cave), and w.r.t. the projection of C_{poly}.

projection of the vectors β^i_{poly} and the set B_{poly} onto the (β_2, β_3)-space which is denoted by $\tilde{\beta}^i$ and $\tilde{B}$, respectively. A combined analysis of Table 5.3 and the subdivision of $\mathbf{R}^2$ w.r.t. the projection of C_{poly} shows that only a fraction of concave-convex-concave functions possess a vertex polyhedral convex envelope. Moreover, only one representative of convex-concave-convex functions is necessary to describe C_{poly}, namely $(\beta^3_{\text{poly}})^\intercal f$.

We compare the standard relaxation R_{Std} of f by the individual convex and concave envelope of f_i, with the basic relaxation R_{Bsc} of f using the derived vectors α^i_{vex} and β^i_{poly}, $i = 1, 2, 3$, i.e., $(x, z) \in R_{\text{Bsc}}$ if and only if $\text{vex}_{[l,u]}[(\alpha^i_{\text{vex}})^\intercal f](x) = (\alpha^i_{\text{vex}})^\intercal f(x) \leq (\alpha^i_{\text{vex}})^\intercal z$ and $\text{vex}_{[l,u]}[(\beta^i_{\text{poly}})^\intercal f](x) \leq$

$(\beta^i_{\text{poly}})^\intercal z$, $i = 1, 2, 3$. This system is equivalent to

$$\begin{array}{rcrcrcrcrcl}
20\,x^3 & - & {}^9\!/_2\,x^5 & +x^6 & \leq & 20\,z_1 & - & {}^9\!/_2\,z_2 & + & z_3, \\
{}^5\!/_{16}\,x^3 & - & {}^9\!/_8\,x^5 & +x^6 & \leq & {}^5\!/_{16}\,z_1 & - & {}^9\!/_8\,z_2 & + & z_3, \\
-2\,x^3 & + & {}^{63}\!/_{20}\,x^5 & -x^6 & \leq & -2\,z_1 & + & {}^{63}\!/_{20}\,z_2 & - & z_3, \\
-{}^{3384}\!/_{167}\,x & + & {}^{1472}\!/_{167} & & \leq & -{}^{1900}\!/_{167}\,z_1 & + & {}^{1287}\!/_{334}\,z_2 & - & z_3, \\
-{}^{21719}\!/_{49682}\,x & + & {}^{11259}\!/_{99241} & & \leq & -{}^{475}\!/_{368}\,z_1 & + & {}^{423}\!/_{184}\,z_2 & - & z_3, \\
{}^{63}\!/_{20}\,x & - & 1 & & \leq & {}^{421}\!/_{80}\,z_1 & - & {}^{63}\!/_{20}\,z_2 & + & z_3.
\end{array}$$

The volumes of the relaxations are calculated with Mathematica 8 [Wol08] and are approximately 589.82 for R_{Std} and 3.51 for R_{Bsc}. Thus, the volume of R_{Std} is more than 147 times larger than the one of R_{Bsc}. ◇

Finally, we emphasize the applicability of our concept to families of interesting functions and domains for which the requirements of Theorems 5.41 and 5.45 hold, e.g.,

- $f = (x^{k_1}, x^{k_2}, x^{k_3})$ with $k_1 < k_2 < k_3$ and $k_i \in \mathbf{R} \setminus [0, 1]$, $i = 1, 2, 3$, and $[l, u] \subseteq \mathbf{R}_{>0}$,
- $f = (\exp(k_1 x), \exp(k_2 x), \exp(k_3 x))$ with $k_3 > k_2 > k_1$ and $k_i \in \mathbf{R}$, and $[l, u] \subseteq \mathbf{R}$,
- $f = (1/x, x^2, \exp(x))$ and $[l, u] \subseteq \mathbf{R}_{>0}$, and
- $f = (-\sqrt{x}, \sin(x), \exp(x))$ and $[l, u] = [3.5, 4.5]$.

In contrast to the vectors of functions discussed in the literature overview in Section 5.1, the vectors presented above are not necessarily complete. For instance, the moment curve consists of all monomials up to a certain degree whereas this completeness is not necessary for our approach. Moreover, while most results in the literature concern vectors of monomials, our approach addresses a more general class of functions.

5.4. Computations

We showed in Examples 5.35 and 5.48 that the volume of the relaxation of a vector of functions can be reduced by orders of magnitude using the proposed basic simultaneous relaxation R_{Bsc} based on α^i_{vex} and β^i_{poly} from Theorems 5.32, 5.41, and 5.45 instead of using the individual convex relaxations of the functions. In this section we give computational evidence for the potential of the proposed relaxations in global optimization software. In Section 5.4.1 we focus on one particular example from

GLOBALLib and present the results of an ad-hoc implementation. In Section 5.4.2 we show results for randomly generated instances which are solved by standard software as well as SCIP [Ach07] with new separators based on the proposed simultaneous relaxations.

5.4.1. Example ex8_4_6 from GLOBALLib

We searched the two common problem libraries GLOBALLib [GLO] and MINLPLib [BDM03] for examples illustrating the effect of simultaneous convexification. Many of the instances in these libraries are rather sparse, that is a lot of variables occur only once or twice so that it is quite unlikely that the simultaneous convexification of several nonlinearities has significant impact. One interesting instance is `ex8_4_6` from GLOBALLib:

$$\min \sum_{i=1}^{8} \left(\frac{x_i - c_i}{x_i} \right)^2 \quad \text{s. t. } x_i = \sum_{j=1}^{3} y_j \exp(-a_i z_j), \quad i = 1, \dots, 8$$

with $x \in [0,1]^8$, $y \in [-10,10]^3$, $z \in [0,0.5]^3$, $a = (4,8,12,24,48,72,94,118)$ and $c = (0.1622, 0.6791, 0.6790, 0.3875, 0.1822, 0.1249, 0.0857, 0.0616)$. The generic lower bound on this problem is 0 and a feasible solution is known with objective function value 0.0011. The problem contains three families of convex functions which all satisfy the conditions of Theorems 5.32, 5.41, and 5.45, namely $f^j(z_j)$, $j = 1,2,3$, with $f_i^j(z_j) := \exp(-a_i z_j)$, $i = 1, \dots, 8$. For instance, f_1^j and f_2^j satisfy condition (ii) of Theorem 5.32: $((f_1^j)'' / (f_2^j)'')' = \exp(4z_j) > 0$.

The proposed relaxations are analyzed in a branch-and-bound framework. For this, standard reformulation and convexification techniques are applied to construct a convex relaxation of the original problem. In particular, each function $f_i^j(z_j) = \exp(-a_i z_j)$ is replaced by a new variable w_i^j.

We investigate several relaxation methods to link the artificial variables w_i^j to the functions f_i^j. First, the standard approach (Stand) is considered corresponding to R_{Std}, where the relaxations are constructed separately for each function. Second, we make use of the basic relaxation R_{Bsc} to relax the simultaneous convex hull of two functions f_r^j and f_s^j, $1 \le r < s \le 8$. For the strength of the relaxation it is important how many pairs of functions f_k^j

and f_l^i are convexified. Relaxation strategy Sim2/4 forms 4 pairs, namely $(k,l) \in \{(1,2),(3,4),(5,6),(7,8)\}$. Relaxation strategy Sim2/7 uses 7 pairs, namely $(k,l) \in \{(1,2),(2,3),(3,4),(4,5),(5,6),(6,7),(7,8)\}$. Third, the basic relaxation for the simultaneous convex hull of three univariate functions is applied analogously. Relaxation strategy Sim3/3 forms 3 triples of functions corresponding to the set $\{(1,2,3),(4,5,6),(6,7,8)\}$ while Sim3/4 considers 4 triples from $\{(1,2,3),(3,4,5),(4,5,6),(6,7,8)\}$.

All relaxation strategies were implemented within a branch-and-bound framework in C++, where each convex subproblem is solved by standard solvers. For this, we applied BARON 11.1.0 [TS05] and CoinBonmin 1.6 [BBC+08]. The package CppAD [CO12] was used to calculate the derivatives of the vectors of univariate functions. The computations were accomplished on a 2.67 GHz INTEL X5650 with 96GB RAM.

A first test of the relaxation strategies showed that none of them could improve the generic lower bound of 0. Therefore, we shrink the domains of the y variables and center them around the best known solution, i.e., $y_1 \in [2.00, 2.10]$, $y_2 \in [0.30, 0.40]$, and $y_3 \in [-4.65, -4.55]$. For this restricted setting Table 5.4 displays the lower bounds and the corresponding number of iterations of the 5 relaxation strategies using BARON after 15, 30, 45, and 60 minutes. The results reveal that the simultaneous convexification of several functions has a significant impact on the overall performance. On the one hand, the convex programs based on the simultaneous convex hulls are more expensive to solve. This is reflected by the fewer number of iterations. For instance, Sim3/4 can only solve a small fraction of convex programs compared to Stand. This can be explained by the high dependency between the variables w_i^j due to the simultaneous convexification. On the other hand, the bounds obtained from simultaneous convexification are often better although less iterations are used.

We also solved the subproblems by the software CoinBonmin which is more suitable for convex problems. The bounds and iterations for Stand obtained with CoinBonmin are similar to the ones computed by BARON. For Sim2/4 and Sim2/7 the bounds by CoinBonmin are worse. Especially for Sim2/4 the lower bound by CoinBonmin after one hour is extremely poor compared to the one of BARON. Note that the severe deviations between the two algorithms are caused by the numerical very unstable problem containing a lot of exponential functions and the different tolerances and limits of the two algorithms. The relaxations Sim3/3 and Sim3/4 based

	Lower Bound / Iterations using BARON			
	15 min	30 min	45 min	60 min
Stand	0.23/1815	0.43/3009	0.61/4272	0.85/5572
Sim2/4	0.05/1116	0.89/2045	1.16/2901	1.37/3782
Sim2/7	7.70/ 992	8.34/1653	8.57/2342	8.76/2971
Sim3/3	0.01/ 496	0.12/1025	6.48/1434	7.38/1760
Sim3/4	0.00/ 47	0.00/ 65	0.00/ 102	0.00/ 120

Table 5.4.: Computations with BARON: Lower bounds scaled by 10^{-4} and number of iterations. The best known feasible solution is $11 \cdot 10^{-4}$.

on the simultaneous convexification of three functions are handled very well by CoinBonmin. In contrast to BARON, the number of iterations of CoinBonmin is enlarged by a factor of 20 which leads to excellent lower bounds. The lower bound of $10.89 \cdot 10^{-4}$ by Sim3/4 after 60 minutes almost proves global optimality of the local solution with objective value $11 \cdot 10^{-4}$.

	Lower Bound / Iterations using CoinBonmin			
	15 min	30 min	45 min	60 min
Stand	0.00/ 37	0.45/3059	0.69/4647	0.90/5873
Sim2/4	0.00/1728	0.03/3590	0.07/5426	0.11/4262
Sim2/7	0.15/ 423	0.42/ 557	3.01/ 823	6.88/1036
Sim3/3	1.46/ 685	7.82/1214	8.57/1960	8.98/2555
Sim3/4	7.69/ 738	9.37/1554	10.37/2226	10.89/2895

Table 5.5.: Computations with CoinBonmin: Lower bounds scaled by 10^{-4} and number of iterations. The best known feasible solution is $11 \cdot 10^{-4}$.

All in all, a relaxation based on the simultaneous relaxation can clearly outperform the standard relaxation. For instance, the lower bounds of Sim2/7 with BARON and Sim3/4 with CoinBonmin are at any time 10 times better than the bound of Stand.

5.4.2. Separators in SCIP

Motivated by the computational results in the previous section, we implemented the proposed basic relaxations R_{Bsc} as separators in SCIP [Ach07]. The separators 2UniVarConv and 3UniVarConv are based on the new relaxations for vectors of two and three univariate convex functions, respectively, given by Theorems 5.32, 5.41, and 5.45. If the corresponding convex constraints $\text{vex}_{[l,u]}[(\alpha^i_{\text{vex}})^\intercal f](x) \le (\alpha^i_{\text{vex}})^\intercal z$ and $\text{vex}_{[l,u]}[(\beta^i_{\text{poly}})^\intercal f](x) \le (\beta^i_{\text{poly}})^\intercal z$ cut off a given point, a linear constraint is generated and added to the linear programming relaxation. In this section we compare the performance of SCIP using the separators with state-of-the-art solvers.

Our test set consists of randomly generated problems, where we investigate the influence of the number of variables `Nvars` and constraints `Ncons`, and the maximum degree `Deg` over all constraints. Similar to Section 3.2.3, we define the following problem class

$$\min\ \epsilon \quad \text{s.t.} \quad \sum_{i=1}^{\texttt{Deg}} \sum_{j=1}^{\texttt{Nvars}} a_{i,j,k}\, x_j^{p_{i,j,k}} \le \epsilon \quad \forall\, k = 1, \ldots, \texttt{Ncons}, \quad x \in [l, u],$$

where $a_{i,j,k}$ is uniformly at random in $\{-4, -3, \ldots, 3, 4\}$, $p_{i,j,k}$ is uniformly at random in $\{i+0.2, i+0.4, \ldots, i+1\}$, l_i is uniformly at random in $\{1, 2, 3, 4, 5\}$, and u_i is uniformly at random in $\{1, 2, 3, 4, 5\} + l_i$ such that $l_i < u_i$. We consider the following parameter settings: $\texttt{Nvars} \in \{10, 20, 30, 50\}$, $\texttt{Ncons} \in \{1, 5, 10, 20, 50\}$, and $\texttt{Deg} \in \{2, 3, 4, 5\}$. For each of the $4 \cdot 5 \cdot 4 = 80$ configurations we generated 10 random instances so that 800 instances are obtained in total.

The functions $x_j^{p_{i,j,k}}$ with the common variable x_j form a vector of univariate convex functions of the form $(x_j^{k_1}, x_j^{k_2}, \ldots, x_j^{k_K})$, $K \in \mathbf{N}$, and $1 < k_r < k_{r+1} \le (\texttt{Deg} + 1)$ for $1 \le r < K$, whose subvectors satisfy the requirements of Theorems 5.32, 5.41, and 5.45. Currently, the separators do not automatically detect these vectors. Instead, we need to reformulate the problems such that each function $x_j^{p_{i,j,k}}$ is replaced by a new variable $w_j^{i,j,k}$ and the constraint $w_j^{i,j,k} = x_j^{p_{i,j,k}}$ is added to the program. This reformulation is called the extended formulation.

We compare SCIP 3.0.0 [Ach07, Ach09] (with the new separators enabled or disabled) with BARON 11.1.0 [TS05] and COUENNE 0.4 [BLL+09]. We use the separators 2UniVarConv and 3UniVarConv individually and jointly, which is denoted by 2-SCIP, 3-SCIP, and 2+3-SCIP. To activate the

separators, we set their frequency to one, i.e., they are applied at every node. All computations were executed on a 2.67 GHz INTEL X5650 with 96GB RAM. The time limit was 15 minutes. Among the 800 instances there are 19 for which one of the algorithms aborted or failed so that they are excluded from our subsequent discussion.

In Table 5.6 we compare the results of the standard solvers BARON, COUENNE, and SCIP for the original and the extended problem formulation. For an explanation of the performance parameters see the discussion of Table 3.2 in Section 3.2.3. The three solvers perform similarly on the

	Original / extended formulation		
	BARON	COUENNE	SCIP
#solved	531 / 549	503 /498	497 / 542
#fastest	103 / 76	7 / 1	47 / **155**
#best dual bound	531 / 549	507 / 504	497 / 542
mean time	64.3 / 60.8	81.4 / 84.8	459.4 / 58.3
mean nodes	265.7 / **138.4**	1249.9 / 1192.6	28395.7 / 823.4
mean dual gap	14.1% / 13.1%	13.9% / 14.2%	14.8% / 10.4

Table 5.6.: Computational results for 800 instances in original and extended formulation.

test set w.r.t. the number of instances solved, the best bound, and the dual gap. BARON's and COUENNE's performance is more or less independent from the problem formulation while SCIP benefits from the extended formulation. It solves 10% more instances and reduces its mean number of nodes and mean computation time significantly from 28395.7 nodes to 823.4 nodes and from 459.4 seconds to 58.3 seconds, respectively. Moreover, SCIP derives the lowest dual gap of 10.48%. For a comparison of the standard solvers with the new separators we thus concentrate on SCIP.

The results in Table 5.7 show that SCIP can take advantage of the separators. SCIP with both separators enabled solves up to 100 additional instances to optimality, derives the best bounds for 737 instances, and even reduces slightly the mean computation time. The strength of the relaxations used by the separators is depicted by the dual gap which is less than 1% compared to 10.48% of SCIP. Comparing the results of the separators, it turns out that all algorithms can solve a comparable num-

	SCIP	2-SCIP	3-SCIP	2+3-SCIP
#solved	542	627	**644**	643
#fastest	**155**	152	86	78
#best dual bound	542	637	708	**737**
mean time	58.3	**44.5**	52.6	53.1
mean nodes	823.4	1031.2	886.0	900.4
mean dual gap	10.48%	**0.52%**	0.76%	**0.49%**

Table 5.7.: SCIP (separators dis-/enabled) applied to 800 instances in extended formulation.

ber of instances. The increasing strength of the relaxations used in 2-SCIP, 3-SCIP, and 2+3-SCIP is reflected by the number of instances for which the algorithms compute the best dual bound, namely 637, 708, and 737. However, the stronger relaxations also lead to slower computations, e.g., 2-SCIP exhibits the lowest mean computation time and is fastest for 152 instances whereas 3-SCIP and 2+3-SCIP are only fastest for 86 and 78 instances, respectively. Nevertheless, the dual bounds of 2+3-SCIP are the strongest so that we focus on this algorithm subsequently.

A detailed analysis of the dual gaps of BARON, COUENNE, SCIP, and 2+3-SCIP w.r.t. the number of variables and constraints, and the highest degree of the polynomials over all constraints is presented in Figure 5.11. The results of the standard solvers are similar: The dual gap increases tremendously for higher number of variables and constraints, and for a higher degree. However, the dual gap decreases slightly when the number of constraints is increased from 20 to 50. In this case the feasible sets might be smaller or potential solutions are excluded more quickly so that the branch-and-bound algorithms become faster. In contrast to the standard solvers, the dual gaps of 2+3-SCIP increase modestly for larger problem instances and none of the gaps is larger than 2%. For the largest problem classes with `Nvars`=50, `Ncons`=50, and `Deg`=5 the algorithm 2+3-SCIP reduces the dual gap of the best standard solver by factors of 30, 10, and 36, respectively.

The overall computations show that SCIP is substantially improved by the implemented separators w.r.t. the considered test set. While SCIP with separators enabled performs similar to the standard solvers for "easier" instances (with a small number of variables and constraints, and a small

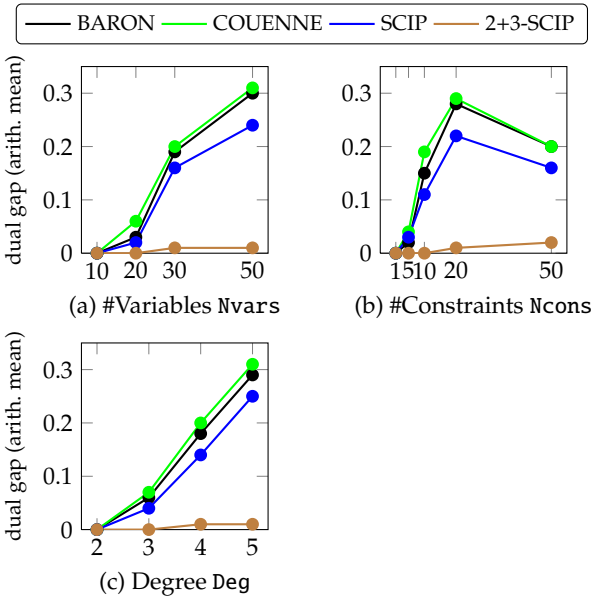


Figure 5.11.: Dual gaps of the solvers w.r.t. the number of variables `NVars`, constraints `Ncons`, and the maximal degree `Deg`.

polynomial degree), it clearly outperforms the state-of-the-art global optimization solvers for the more complicated instances. For these instances there is a high dependence between the different univariate convex functions so that the separators can significantly exploit the simultaneous relaxations.

In this chapter we established a link between the simultaneous convex hull $Q_D[f]$ of $f = (f_1, \dots, f_m)$ and the convex envelope of the functions $\alpha^\top f$, $\alpha \in \mathbf{R}^m$, in order to use the theory of convex envelopes to derive properties for $Q_D[f]$. On the one hand, we showed that the union of the generating sets of $\mathrm{vex}_D[\alpha^\top f]$ over all $\alpha \in \mathbf{R}^m$ is dense in the generating set of $Q_D[f]$. On the other hand, we described $Q_D[f]$ via the convex envelopes of $\alpha^\top f, \alpha \in \mathbf{R}^m$, and identify subsets of $\alpha \in \mathbf{R}^m$ which are not necessary for this representation, namely the interior of the cones C_{vex} and C_{poly}. Based on this, a strong relaxation of $Q_D[f]$ for vectors of two and three univariate convex functions was proposed. Computational results showed that the proposed simultaneous relaxations do not only yield theoretically better relaxations but also accelerate global optimization software.

Outlook

This thesis presented new techniques to construct strong convex relaxations of MINLPs. The impact of these techniques was demonstrated in various case studies involving benchmark problems as well as two applications from chemical engineering. Nevertheless, the results presented in this thesis can only be viewed as small steps towards the global optimization of general real-world applications. In each part of the thesis interesting further research directions can be inferred to accelerate the computations of MINLPs.

In Chapter 2 we deduced a bound tightening technique for a hybrid separation process of a binary mixture. Motivated by the good computational results, an extension of this technique handling processes with multicomponent mixtures could be investigated. In this multivariate setting the interaction between the variables increases so that the analysis of the equation system becomes more complicated.

In Chapter 3 strong convex relaxations for two classes of bivariate functions are discussed. For bivariate functions with a fixed convexity behavior we implemented a constraint handler in SCIP which automatically detects the convexity behavior of bivariate quadratic and monomial functions. Each function is then replaced by a new variable and individually handled by the constraint handler. Hence, it is not exploited that the sum of functions, which exhibit the same fixed convexity pattern, is again characterized by this convexity pattern. Using this fact, the sum of functions can be treated as one function and less additional variables need to be introduced and relaxed by the constraint handler.

In Chapter 4 we derived extended formulations for the convex envelope

of practically relevant functions based on a simultaneous convexification with multilinear monomials. For this, we introduce all possible multilinear monomials of a certain family of variables so that the RLT theory can be used to uniquely solve the corresponding optimization problem. However, this leads to an exponential increase in the number of additional variables. It is thus of interest to investigate the convex envelope in a reduced, extended space, corresponding to a subset of the multilinear monomials, which yet enables us to derive closed-form expressions for the convex envelope.

In Chapter 5 both an inner and outer description for the simultaneous convex hull of several functions is provided based on convex envelopes. Regarding the inner description we proved that the generators of the simultaneous convex hull are dense in the generators of the convex envelopes, but we could not find any example which gives evidence for the gap between the two sets of generators. Regarding the outer description we suggested a strong basic relaxation for the simultaneous convex hull of two and three univariate convex functions, which is derived by the explicit use of analytical tools. For more functions this approach is not appropriate as the analytical solution of the corresponding auxiliary problems becomes more and more complicated. Alternative approaches are therefore necessary that might deliver weaker results for two and three functions but are applicable to larger families of functions.

APPENDIX A

Modifications of (S1) and (S2)

Test instance	Optimal cost	Lower bounds and CPU time (hh:mm) (S1mod)	(S2mod)
T0	**306.37**	252.97 (100:00)	202.65 (100:00)
T2a	**254.33**	201.64 (100:00)	218.81 (100:00)
T2b	**326.17**	250.13 (100:00)	222.92 (100:00)
T4a	**153.17**	115.18 (100:00)	108.28 (100:00)
T4b	**612.77**	497.12 (100:00)	512.83 (100:00)
T5a	**282.00**	253.02 (100:00)	203.37 (100:00)
T5b	**531.10**	415.11 (100:00)	427.96 (100:00)
T6a	**106.20**	78.32 (100:00)	48.44 (100:00)
T6b	**175.17**	124.57 (100:00)	89.20 (100:00)
T7a	**1056.47**	224.68 (100:00)	115.27 (100:00)
T7b	**155.50**	155.50 (52:08)	135.84 (100:00)

Table A.1.: Stand-alone distillation column: Optimal cost in comparison with the lower bounds of (S1mod) and (S2mod) after 100 hours or the time needed to solve the problem globally. In (S1mod) the reference model is changed such that nonlinearities including a binary variable are removed using a Big-M formulation. In (S2mod) the modeling approach of [VN11] is used to reduce the number of binary variables.

APPENDIX B

Copyrights

The permission to reproduce certain copyright material was obtained by license agreements - between the author of this thesis and the corresponding publishers - provided by the Copyright Clearance Center.

- Figure 3.3 is a reprint of Figure 1 in the paper: *A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs*, Jeff Linderoth, Mathematical Programming **103** (2005), no. 2, 251–282, with kind permission from Springer Science and Business Media.

- Section 3.3 is based on the paper: *A theoretical study of continuous counter-current chromatography for adsorption isotherms with inflection points*, Martin Ballerstein, Dennis Michaels, Andreas Seidel-Morgenstern, and Robert Weismantel, Computers & Chemical Engineering **34** (2010), no. 4, 447–459, with permission from Elsevier.

- Figures 3.10 (a) and (b) are reprints of Figures 5 and 2, respectively, in : *Continuous chromatographic separation through simulated moving beds under linear and nonlinear conditions*, Cristiano Migliorini, Marco Mazzotti, and Massimo Morbidelli, Journal of Chromatography A **827** (1998), 161–173, with permission from Elsevier.

- Figure 3.10 (c) is a reprint of Figure 3 in: *Design of simulated moving bed processes under reduced purity requirements*, Malte Kaspereit,

Bibliography

[AB06] Charalambos D. Aliprantis and Kim C. Border, *Convexity*, Infinite Dimensional Analysis, Springer Berlin Heidelberg, 2006, pp. 251–309.

[AB10] Kurt Anstreicher and Samuel Burer, *Computable representations for convex hulls of low-dimensional quadratic forms*, Mathematical Programming **124** (2010), no. 1-2, 33–43.

[ABH12] Tobias Achterberg, Timo Berthold, and Gregor Hendel, *Rounding and propagation heuristics for mixed integer programming*, Operations Research Proceedings 2011 (Diethard Klatte, Hans-Jakob Lüthi, and Karl Schmedders, eds.), Springer Berlin Heidelberg, 2012, pp. 71–76.

[Ach07] Tobias Achterberg, *Constraint Integer Programming*, Ph.D. thesis, Technische Universität Berlin, 2007.

[Ach09] ———, *SCIP: Solving Constraint Integer Programs*, Mathematical Programming Computation **1** (2009), no. 1, 1–41.

[ADFN98] Claire S. Adjiman, S. Dallwig, Christodoulos A. Floudas, and Arnold Neumaier, *A global optimization method, αBB, for general twice-differentiable constrained NLPs – I. Theoretical advances*, Computers & Chemical Engineering **22** (1998), no. 9, 1137–1158.

[AF04a] Ionnis G. Akrotirianakis and Christodoulos A. Floudas, *A New Class of Improved Convex Underestimators for Twice Continuously Differentiable Constrained NLPs*, Journal of Global Optimization **30** (2004), no. 4, 367–390.

[AF04b] _____, *Computational Experience with a New Class of Convex Underestimators: Box-constrained NLP Problems*, Journal of Global Optimization **29** (2004), no. 3, 249–264.

[Ans09] Kurt M. Anstreicher, *Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming*, Journal of Global Optimization **43** (2009), no. 2-3, 471–484.

[AS05] Warren P. Adams and Hanif D. Sherali, *A hierarchy of relaxations leading to the convex hull representation for general discrete optimization problems*, Annals of Operations Research **140** (2005), no. 1, 21–47.

[BA02] Mariana Barttfeld and Pío A. Aguirre, *Optimal synthesis of multicomponent zeotropic distillation processes. 1. Preprocessing phase and rigorous optimization for a single unit*, Industrial & Engineering Chemistry Research **41** (2002), no. 21, 5298–5307.

[BA03] _____, *Optimal synthesis of multicomponent zeotropic distillation processes. 2. Preprocessing phase and rigorous optimization of efficient sequences*, Industrial & Engineering Chemistry Research **42** (2003), no. 14, 3441–3457.

[Bal08] Martin Ballerstein, *Relaxation strategies for statistical isotherms of second degree in 4-zone tmb processes*, 2008, Diploma thesis, Otto-von-Guericke Universität Magdeburg.

[BBC+08] Pierre Bonami, Lorenz T. Biegler, Andrew R. Conn, Gérard Cornuéjols, Ignacio E. Grossmann, Carl D. Laird, Jon Lee, Andrea Lodi, Francois Margot, Nicolas Sawaya, and Andreas Wächter, *An algorithmic framework for convex mixed integer nonlinear programs*, Discrete Optimization **5** (2008), no. 2, 186 – 204.

[BCLL12] Pietro Belotti, Sonia Cafieri, Jon Lee, and Leo Liberti, *On feasibility based bound tightening*, available at `http://www.optimization-online.org/DB_FILE/2012/01/3325.pdf`, 2012.

[BDH96] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa, *The quickhull algorithm for convex hulls*, ACM Transactions on Mathematical Software **22** (1996), no. 4, 469–483.

[BDM03] Michael R. Bussieck, Arne Stolbjerg Drud, and Alexander Meeraus, *MINLPLib - a collection of test models for mixed-integer nonlinear programming*, INFORMS Journal on Computing **15** (2003), no. 1, 114–119.

[Ben04] Harold P. Benson, *On the construction of convex and concave envelope formulas for bilinear and fractional functions on quadrilaterals*, Computational Optimization and Applications **27** (2004), no. 1, 5–22.

[BGGP99] Frédéric Benhamou, Frédéric Goualard, Laurent Granvilliers, and Jean-Francois Puget, *Revising hull and box consistency*, ICLP, 1999, pp. 230–244.

[BGSA08] María Lorena Bergamini, Ignacio Grossmann, Nicolás Scenna, and Pío Aguirre, *An improved piecewise outer-approximation algorithm for the global optimization of MINLP models involving concave and bilinear terms*, Computers & Chemical Engineering **32** (2008), no. 3, 477–493.

[BHSM03] Daria Beltscheva, Peter Hugo, and Andreas Seidel-Morgenstern, *Linear two-step gradient counter-current chromatography analysis based on a recursive solution of an equilibrium stage model*, Journal of Chromatography A **989** (2003), no. 1, 31–45.

[BHV09] Timo Berthold, Stefan Heinz, and Stefan Vigerske, *Extending a CIP framework to solve MIQCPs*, Mixed-integer nonlinear optimization: Algorithmic advances and applications (Jon Lee and Sven Leyffer, eds.), IMA volumes in Mathematics and its Applications, vol. 154, Springer, 2009, pp. 427–444.

[BKK+] Martin Ballerstein, Achim Kienle, Christian Kunde, Dennis Michaels, and Robert Weismantel, *Deterministic global optimization of binary hybrid distillation/melt-crystallization processes based on relaxed MINLP formulations*, submitted, 2012.

[BKK+11] ———, *Towards global optimization of combined distillation-crystallization processes for the separation of closely boiling mixtures*, 21th European Symposium on Computer Aided Process Engineering - ESCAPE 21 (Efstratios N. Pistikopoulos, Michael C. Georgiadis, and Antonis C. Kokossis, eds.), Elsevier, 2011, pp. 552–556.

[BL09] Samuel Burer and Adam N. Letchford, *On nonconvex quadratic programming with box constraints*, SIAM Journal on Optimization **20** (2009), no. 2, 1073–1089.

[BL12] ———, *Non-convex mixed-integer nonlinear programming: A survey*, Surveys in Operations Research and Management Science **17** (2012), no. 2, 97 – 106.

[BLL+09] Pietro Belotti, Jon Lee, Leo Liberti, François Margot, and Andreas Wächter, *Branching and bounds tightening techniques for non-convex MINLP*, Optimization Methods and Software **24** (2009), 597–634.

[BM] Martin Ballerstein and Dennis Michaels, *Extended formulations for convex envelopes*, Journal on Global Optimization, accepted, August 2013.

[BMSMW10] Martin Ballerstein, Dennis Michaels, Andreas Seidel-Morgenstern, and Robert Weismantel, *A theoretical study of continuous counter-current chromatography for adsorption isotherms with inflection points*, Computers & Chemical Engineering **34** (2010), no. 4, 447–459.

[BMV13] Martin Ballerstein, Dennis Michaels, and Stefan Vigerske, *Linear underestimators for bivariate functions with a fixed convexity behavior*, Tech. Report 13-02, Zuse Institute Berlin, Takustr. 7, 14195 Berlin, 2013.

[BS12] Samuel Burer and Anureet Saxena, *The milp road to miqcp*, Mixed Integer Nonlinear Programming (Jon Lee and Sven Leyffer, eds.), The IMA Volumes in Mathematics and its Applications, vol. 154, Springer New York, 2012, pp. 373–405.

[BST09] Xiaowei Bao, Nikolaos V. Sahinidis, and Mohit Tawarmalani, *Multiterm polyhedral relaxations for nonconvex, quadratically constrained quadratic programs*, Optimization Methods Software **24** (2009), no. 4-5, 485–504.

[BST11] _____, *Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons*, Mathematical Programming **129** (2011), no. 1, 129–157.

[BWM98] Jürgen Bausa, Rüediger von Watzdorf, and Wolfgang Marquardt, *Shortcut methods for nonideal multicomponent distillation: 1. Simple columns*, AIChE Journal **44** (1998), no. 10, 2181–2198.

[Cah12] Jim Cahill, *Reducing distillation column energy usage*, available at `http://www.emersonprocessxperts.com/2010/04/reducing_distil/`, 2012.

[CCHU92] Chi B. Ching, K. H. Chu, Kus Hidajat, and Mohammed S. Uddin, *Comparative study of flow schemes for a simulated countercurrent adsorption separation process*, AIChE Journal **38** (1992), no. 11, 1744–1750.

[CL07] Thomas Christof and Andreas Löbel, *Porta – POlyhedron Representation Transformation Algorithm*, available at `http://typo.zib.de/opt-long_projects/Software/Porta/`, 2007.

[CL10] Alberto Caprara and Marco Locatelli, *Global optimization problems and domain reduction strategies*, Mathematical Programming **125** (2010), no. 1, 123–137.

[CLL10] Sonia Cafieri, Jon Lee, and Leo Liberti, *On convex relaxations of quadrilinear terms*, Journal on Global Optimization **47** (2010), no. 4, 661–685.

[CO12] COIN-OR, *CppAD*, 2012, available at `http://www.coin-or.org/CppAD/`.

[CPW00] Michael Cook, Colin F. Poole, and Ian D. Wilson (eds.), *Encyclopedia of separation science*, San Diego, Academic Press, 2000.

[DG86] Marco A. Duran and Ignacio E. Grossmann, *An outer-approximation algorithm for a class of mixed-integer nonlinear programs*, Mathematical Programming **36** (1986), no. 3, 307–339.

[DG91] Moustapha Diack and Georges Guiochon, *Adsorption isotherm and overloaded elution profiles of phenyldodecane on porous carbon in liquid chromatography*, Analytical Chemistry **63** (1991), no. 22, 2608–2613.

[DN10] Ferenc Domes and Arnold Neumaier, *Constraint propagation on quadratic constraints*, Constraints **15** (2010), no. 3, 404–429.

[FK] Dietrich Flockerzi and Christian Kunde, personal communication.

[Flo95] Christodoulos A. Floudas (ed.), *Nonlinear and mixed-integer optimization : Fundamentals and applications*, Oxford University Press, New York, 1995.

[FNN+08] Meik Bernhard Franke, Norman Nowotny, Eugene Ndocko Ndocko, Andrzej Gorak, and Jochen Strube, *Design and optimization of a hybrid distillation/melt crystallization process*, AIChE Journal **54** (2008), no. 11, 2925–2942.

[FS69] James E. Falk and Richard M. Soland, *An algorithm for separable nonconvex programming problems*, Management Science **15** (1969), no. 9, 550–569.

[GAB05] Ignacio E. Grossmann, Pío A. Aguirre, and Mariana Barttfeld, *Optimal synthesis of complex distillation columns using rigorous models*, Computers & Chemical Engineering **29** (2005), no. 6, 1203–1215.

[GAM09] GAMS Development Corp., *Gams - the solver manuals*, 2009.

[GFSK06] Georges Guiochon, Attila Felinger, Dean G. Shirazi, and Anita M. Katti, *Fundamentals of preparative and nonlinear chromatography*, 2nd ed., Elsevier, Amsterdam, 2006.

[GHJ+08a] Jignesh Gangadwala, Utz-Uwe Haus, Matthias Jach, Achim Kienle, Dennis Michaels, and Robert Weismantel, *Global analysis of combined reaction distillation processes*, Computers & Chemical Engineering **32** (2008), no. 1-2, 343–355.

[GHJ+08b] Jignesh Gangadwala, Utz-Uwe Haus, Matthias Jach, Achim Kienle, Dennis Michaels, and Robert Weismantel, *Global analysis of combined reaction distillation processes*, Computers & Chemical Engineering **32** (2008), no. 1-2, 343–355.

[GJ00] Ewgenij Gawrilow and Michael Joswig, *polymake: a framework for analyzing convex polytopes*, Polytopes — Combinatorics and Computation (Gil Kalai and Günter M. Ziegler, eds.), Birkhäuser, 2000, pp. 43–74.

[GKH+06] Jignesh Gangadwala, Achim Kienle, Utz-Uwe Haus, Dennis Michaels, and Robert Weismantel, *Global bounds on optimal solutions for the production of 2,3-dimethylbutene-1*, Industrial & Engineering Chemistry Research **45** (2006), no. 7, 2261–2271.

[GLO] GLOBAL Library, `http://www.gamsworld.org/global/globallib.htm`.

[GMNS60] Charles H. Giles, T. H. MacEwan, S. N. Nakhwa, and D. Smith, *Studies in adsorption. part xi. a system of classification of solution adsorption isotherms, and its use in diagnosis of adsorption mechanism and in measurement of specific surface areas of solids.*, Journal of the Chemical Society (1960), 3973–3993.

[GMS06] Philip E. Gill, Walter Murray, and Michael A. Saunders, *User's guide for SNOPT Version 7: Software for large-scale nonlinear programming*, 2006, Available at `http://www.sbsi-sol-optimize.com/manuals/SNOPT\%20Manual.pdf`.

[Hil60] Terrell L. Hill, *Introduction to statistical thermodynamics*, Addison-Wesley, 1960.

[HMSMW07] Utz-Uwe Haus, Dennis Michaels, Andreas Seidel-Morgenstern, and Robert Weismantel, *A method to evaluate the feasibility of TMB chromatography for reduced efficiency and purity requirements based on discrete optimization*, Computers & Chemical Engineering **31** (2007), no. 11, 1525–1534.

[HSYY08] Peter Huggins, Bernd Sturmfels, Josephine Yu, and Debbie Yuster, *The hyperdeterminant and triangulations of the 4-cube*, Mathematics of Computation **77** (2008), no. 263, 1653–1679.

[HT96] Reiner Horst and Hoang Tuy, *Global optimization*, 3rd ed., Springer, 1996.

[HUL01] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal, *Fundamentals of convex analysis*, Springer Berlin Heidelberg, 2001.

[IBM12] IBM, *ILOG CPLEX*, 2009–2012, `http://www.ibm.com/software/integration/optimization/cplex`.

[ILO07] ILOG, *CPLEX*, 1997–2007, `http://www.ilog.com/products/cplex/`.

[JKMW08] Matthias Jach, Achim Kienle, Dennis Michaels, and Robert Weismantel, *Novel convex underestimators and their application to the synthesis of combined reaction distillation processes*, 18th European Symposium on Computer-Aided Process Engineering, Computer-aided Chemical Engineering, Elsevier, 2008.

[JMW08] Matthias Jach, Dennis Michaels, and Robert Weismantel, *The convex envelope of $(n-1)$-convex functions*, SIAM Journal on Optimization **19** (2008), no. 3, 1451–1466.

[KCR98] Richard Kelsey, William Clinger, and Jonathan Rees, *Revised^5 report on the algorithmic language Scheme*, ACM SIGPLAN Notices **33** (1998), no. 9, 26–76.

[Kea06] R. Baker Kearfott, *Discussion and empirical comparisons of linear relaxations and alternate techniques in validated deterministic global optimization*, Optimization Methods and Software **21** (2006), no. 5, 715–731.

[KKM09] Korbinian Krämer, Sven Kossack, and Wolfgang Marquardt, *Efficient optimization-based design of distillation processes for homogenous azeotropic mixtures*, Industrial & Engineering Chemistry Research **48** (2009), no. 14, 6749–6764.

[KS53] Samuel Karlin and Llyod S. Shapley, *Geometry of moment spaces*, Memoirs of the American Mathematical Society, no. 12, American Mathematical Society, Providence, R.I., 1953.

[KS12a] Aida Khajavirad and Nikolaos Sahinidis, *Convex envelopes generated from finitely many compact convex sets*, Mathematical Programming **137** (2012), no. 1-2, 371–408.

[KS12b] ———, *Convex envelopes of products of convex and componentwise concave functions*, Journal of Global Optimization **52** (2012), no. 3, 391–409.

[KSMK07] Malte Kaspereit, Andreas Seidel-Morgenstern, and Achim Kienle, *Design of simulated moving bed processes under reduced purity requirements*, Journal of Chromatography A **1162** (2007), no. 1, 2–13.

[LAT08] Angelo Lucia, Amit Amale, and Ross Taylor, *Distillation pinch points and more*, Computers and Chemical Engineering **32** (2008), no. 6, 1350–1372.

[Lau03] Monique Laurent, *A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0-1 programming*, Mathematics of Operations Research **28** (2003), no. 3, 470–496.

[Lav06] Carlile Lavor, *On generating instances for the molecular distance geometry problem*, Global Optimization. From Theory to Implementation (Leo Liberti and Nelson Maculan, eds.), Springer, 2006, pp. 405–414.

[Lin05] Jeff Linderoth, *A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs*, Mathematical Programming **103** (2005), no. 2, 251–282.

[LLM09] Carlile Lavor, Leo Liberti, and Nelson Maculan, *Molecular distance geometry problem*, Encyclopedia of Optimization (Christodoulos A. Floudas and Panos M. Pardalos, eds.), Springer, 2nd ed., 2009, pp. 2304–2311.

[LMR05] Yahia Lebbah, Claude Michel, and Michel Rueher, *A rigorous global filtering algorithm for quadratic constraints*, Constraints **10** (2005), no. 1, 47–65.

[Loc10] Marco Locatelli, *Convex envelopes for quadratic and polynomial functions over polytopes*, available at `www.optimization-online.org/DB_FILE/2010/11/2788.pdf`, 2010.

[LP03] Leo Liberti and Constantinos C. Pantelides, *Convex envelopes of monomials of odd degree*, Journal of Global Optimization **25** (2003), no. 2, 157–168.

[LS10] Marco Locatelli and Fabio Schoen, *On convex envelopes and underestimators for bivariate functions*, available at `www.optimization-online.org/DB_FILE/2009/11/2462.pdf`, 2010.

[LVD85] Sanford G. Levy, David B. Van Dongen, and Michael F. Doherty, *Design and synthesis of homogeneous azeotropic distillations. 2. Minimum reflux calculations for nonideal and azeotropic columns*, Industrial & Engineering Chemistry Fundamentals **24** (1985), no. 4, 463–474.

[Maz06] Marco Mazzotti, *Design of simulated moving bed separations: Generalized Langmuir isotherm*, Industrial & Engineering Chemistry Research **45** (2006), no. 18, 6311–6324.

[MBR+11] Jovana Micovic, Thorsten Beierling, Felly Ruether, Peter Kreis, and Andrzej Górak, *Hybrid separation processes for purification of close boiling mixtures in hydroformylation of long chain olefins*, 8th European Congress of Chemical Engineering (ECCE-8), 2011.

[McC76] Garth P. McCormick, *Computability of global solutions to factorable nonconvex programs. I: Convex underestimating problems*, Mathematical Programming **10** (1976), no. 1, 147–175.

[Mes04] Frédéric Messine, *Deterministic global optimization using interval constraint propagation techniques*, RAIRO - Operations Research **38** (2004), no. 4, 277–293.

[MF94] Costas D. Maranas and Christodoulos A. Floudas, *Global minimum potential energy conformations of small molecules*, Journal of Global Optimization **4** (1994), no. 2, 135–170.

[MF95] _____, *Finding all solutions of nonlinearity constrained systems of equations*, Journal of Global Optimization **7** (1995), no. 2, 143–182.

[MF03] Clifford A. Meyer and Christodoulos A. Floudas, *Trilinear monomials with positive or negative domains: Facets of the convex and concave envelopes*, Frontiers in Global Optimization (Christodoulos A. Floudas and Panos M. Pardalos, eds.), Kluwer Academic Publishers, 2003, pp. 327–352.

[MF04] Clifford A. Meyer and Christodoulos A. Floudas, *Trilinear monomials with mixed sign domains: Facets of the convex and concave envelopes*, Journal of Global Optimization **29** (2004), 125–155.

[MF05] Clifford A. Meyer and Christodoulos A. Floudas, *Convex envelopes for edge-concave functions*, Mathematical Programming **103** (2005), no. 2, 207–224.

[Mic07] Dennis Michaels, *Discrete optimization techniques for nonlinear mixed-integer optimization problems arising from chemical engineering*, Der Andere Verlag, Tönning, 2007.

[MMM98] Cristiano Migliorini, Marco Mazzotti, and Massimo Morbidelli, *Continuous chromatographic separation through simulated moving beds under linear and nonlinear conditions*, Journal of Chromatography A **827** (1998), 161–173.

[Moo66] Ramon E. Moore, *Interval analysis*, Prentince-Hall, Englewood Cliffs, USA, 1966.

[MSMG04] Kathleen Mihlbachler, Andreas Seidel-Morgenstern, and Georges Guiochon, *Detailed study of Tröger's base enantiomers by SMB processes*, American Institute of Chemical Engineers **40** (2004), no. 3, 611–623.

[Pad75] Manfred W. Padberg, *A note on 0/1 programming*, Operations Research **23** (1975), no. 4, 833–837.

[Pou13] Lionel Pournin, *The flip-graph of the 4-dimensional cube is connected*, Discrete & Computational Geometry **49** (2013), no. 3, 511–530.

[RC89] Douglas M. Ruthven and C. B. Ching, *Counter-current and simulated counter-current adsorption separation processes*, Chem Eng Sci. **44** (1989), 1011–1038.

[Rik97] Anatoliy D. Rikun, *A convex envelope formula for multilinear functions*, Journal of Global Optimization **10** (1997), no. 4, 425–437.

[Roc70] R. Tyrrell Rockafellar, *Convex analysis*, Princeton University Press, Princeton, New Jersey, 1970.

[RPM09] Arvind Rajendran, Galatea Paredes, and Marco Mazzotti, *Simulated moving bed chromatography for the separation of enantiomers*, Journal of Chromatography A **1216** (2009), no. 4, 709–738.

[RS95] Hong S. Ryoo and Nikolaos V. Sahinidis, *Global optimization of nonconvex nlps and minlps with applications in process design*, Computers & Chemical Engineering **19** (1995), no. 5, 551–566.

[RS96] Hong S. Ryoo and Nikolaos V. Sahinidis, *A branch-and-reduce approach to global optimization*, Journal of Global Optimization **8** (1996), no. 2, 107–138.

[RT10] Jean-Philippe P. Richard and Mohit Tawarmalani, *Lifting inequalities: a framework for generating strong cuts for nonlinear programs*, Mathematical Programming **121** (2010), no. 1, 61–104.

[SA90] Hanif D. Sherali and Warren P. Adams, *A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems*, SIAM Journal of Discrete Mathematics **3** (1990), no. 3, 411–430.

[SA94] ———, *A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems*, Discrete Applied Mathematics **52** (1994), no. 1, 83–106.

[SA09] ———, *A reformulation-linearization technique (RLT) for semi-infinite and convex programs under mixed 0-1 and general discrete restrictions*, Discrete Applied Mathematics **157** (2009), no. 6, 1319–1333.

[Sah03] Nikolaos Sahinidis, *Global optimization and constraint satisfaction: The branch-and-reduce approach*, Global Optimization and Constraint Satisfaction (Christian Bliek, Christophe Jermann, and Arnold Neumaier, eds.), Lecture Notes in Computer Science, vol. 2861, Springer Berlin / Heidelberg, 2003, pp. 1–16.

[SDD12] Hanif D. Sherali, Evrim Dalkiran, and Jitamitra Desai, *Enhancing RLT-based relaxations for polynomial programming problems via a new class of v-semidefinite cuts*, Computational Optimization and Applications **52** (2012), no. 2, 483–506.

[SDL12] Hanif D. Sherali, Evrim Dalkiran, and Leo Liberti, *Reduced RLT representations for nonconvex polynomial programming problems*, Journal of Global Optimization **52** (2012), no. 3, 447–469.

[Sel70] Samuel M. Selby (ed.), *Standard mathematical tables*, 18th ed., The Chemical Rubber Company, Cleveland, Ohio, 1970.

[She97] Hanif D. Sherali, *Convex envelopes of multilinear functions over a unit hypercube and over special sets*, Acta Mathematica Vietnamica **22** (1997), no. 1, 245–270.

[She98] Hanif D. Sherali, *Global optimization of nonconvex polynomial programming problems having rational exponents*, Journal of Global Optimization **12** (1998), no. 3, 267–283.

[SMMC93] Giuseppe Storti, Marco Mazzotti, Massimo Morbidelli, and Sergio Carra, *Robust design of binary countercurrent adsorption separation process*, AIChE Journal **39** (1993), no. 3, 471–492.

[ST92] Hanif D. Sherali and Cihan H. Tuncbilek, *A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique*, Journal of Global Optimization **2** (1992), no. 1, 101–112.

[ST97] Hanif D. Sherali and Cihan H. Tuncbilek, *New reformulation linearization/convexification relaxations for univariate and multivariate polynomial programming problems*, Operations Research Letters **21** (1997), no. 1, 1–9.

[SW01] Hanif D. Sherali and Hongjie Wang, *Global optimization of nonconvex factorable programming problems*, Mathematical Programming **89** (2001), no. 3, 459–478.

[Tar03] Fabio Tardella, *On the existence of polyedral convex envelopes*, Frontiers in Global Optimization, Kluwer Academic Publisher, 2003, pp. 563–573.

[Tar08] ———, *Existence and sum decomposition of vertex polyhedral convex envelopes*, Optimization Letters **2** (2008), no. 3, 363–375.

[Taw10] Mohit Tawarmalani, *Inclusion certificates and simultaneous convexification of functions*, available at `http://www.optimization-online.org/DB_FILE/2010/09/2722.pdf`, 2010.

[TRX12] Mohit Tawarmalani, Jean-Philippe P. Richard, and Chuanhui Xiong, *Explicit convex and concave envelopes through polyhedral subdivisions*, Mathematical Programming (2012), 1–47.

[TS01] Mohit Tawarmalani and Nikolaos V. Sahinidis, *Semidefinite relaxations of fractional programs via novel convexification techniques*, Journal of Global Optimization **20** (2001), no. 2, 137–158.

[TS02a] _____, *Convex extensions and envelopes of lower semi-continuous functions*, Mathematical Programming **93** (2002), no. 2, 247–263.

[TS02b] Mohit Tawarmalani and Nikolaos V. Sahinidis, *Convexification and global optimization in continuous and mixed-integer nonlinear programming: Theory, algorithms, software, and applications*, Nonconvex Optimization And Its Applications, vol. 65, Kluwer Academic Publishers, 2002.

[TS04] Mohit Tawarmalani and Nikolaos V. Sahinidis, *Global optimization of mixed-integer nonlinear programs: A theoretical and computational study*, Mathematical Programming **99** (2004), no. 3, 563–591.

[TS05] _____, *A polyhedral branch-and-cut approach to global optimization*, Mathematical Programming **103** (2005), no. 2, 225–249.

[VG90] Jagadisan Viswanathan and Ignacio E. Grossmann, *A combined penalty function and outer-approximation method for MINLP optimization*, Computers & Chemical Engineering **14** (1990), no. 7, 769–782.

[VG93] _____, *Optimal feed locations and number of trays for distillation columns with multiple feeds*, Industrial & Engineering Chemistry Research **32** (1993), no. 11, 2942–2949.

[Vig12] Stefan Vigerske, *Decomposition in multistage stochastic programming and a constraint integer programming approach to mixed-integer nonlinear programming*, Ph.D. thesis, Humboldt-Universität Berlin, 2012, to be published.

[VN11] Juan Vielma and George Nemhauser, *Modeling disjunctive constraints with a logarithmic number of binary variables and constraints*, Mathematical Programming **128** (2011), no. 1, 49–72.

[Wal04] Wolfgang Walter, *Analysis 1*, 7th ed., Springer Berlin, 2004.

[Wol08] Wolfram Research, Inc., *Mathematica*, 2008.

[YG00] Hector Yeomans and Ignacio E. Grossmann, *Disjunctive programming models for the optimal design of distillation columns and separation sequences*, Industrial & Engineering Chemistry Research **39** (2000), no. 6, 1637–1648.

[ZSSM06] Weibing Zhang, Yichu Shan, and Andreas Seidel-Morgenstern, *Breakthrough curves and elution profiles of single solutes in case of adsorption isotherms with two inflection points*, Journal of Chromathography A **1107** (2006), no. 1-2, 216–225.

List of Figures

List of Tables