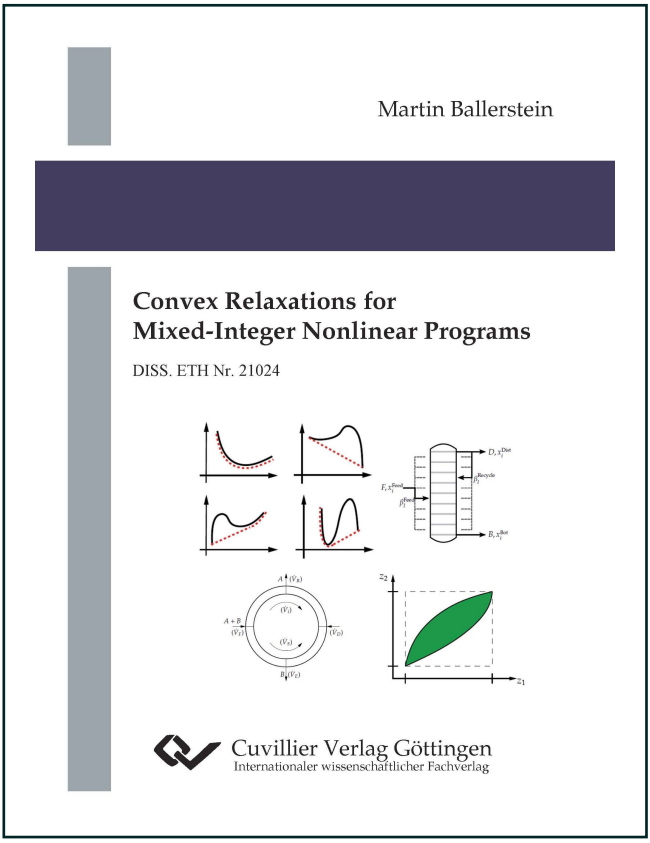
Martin Ballerstein (Autor)

Convex Relaxations for Mixed-Integer Nonlinear Programs



https://cuvillier.de/de/shop/publications/6445

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

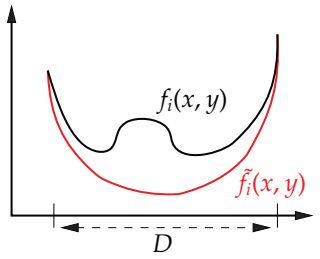
convex relaxations. While the local solvers determine a feasible solution whose objective function value constitutes an upper bound on the problem, the solution of the convex relaxation corresponds to a lower bound. Consequently, an optimal solution is found if the two bounds coincide.

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].

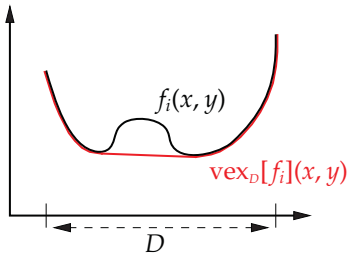
All deterministic algorithms have in common that their convergence heavily depends on the strength of the convex relaxations. For example, stronger relaxations allow to detect infeasibility of a node in the branch-and-bound algorithm more easily so that the exploration of further child nodes can be avoided. 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 $\text{vex}_D[f]$. 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

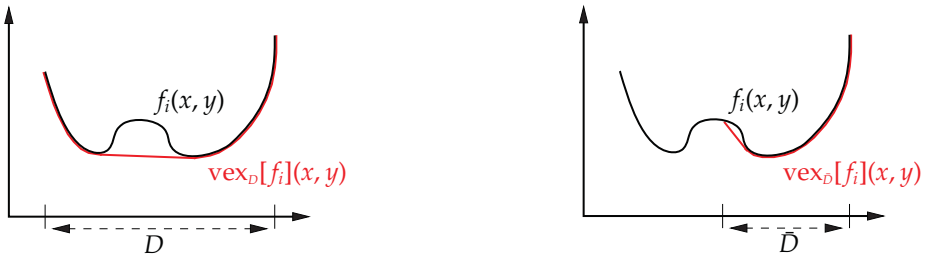
$$h_1 = h_2 h_3, \qquad h_2 = h_4^2, \qquad h_3 = \frac{1}{1+\exp(h_4)}, \qquad h_4 = x_1 x_2.$$

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 underestimators 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 $f(x) = {}^{x_1x_2}/_{x_3}$, $x_1 \in [-1, 1], x_2 \in [0.1, 1], x_3 \in [0.1, 1]$. 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.

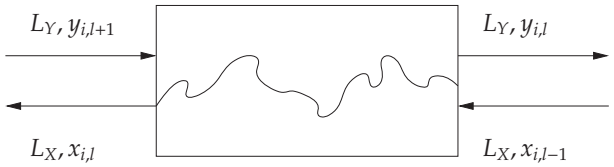


Figure 2.1.: Principle of material balance equations.

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.

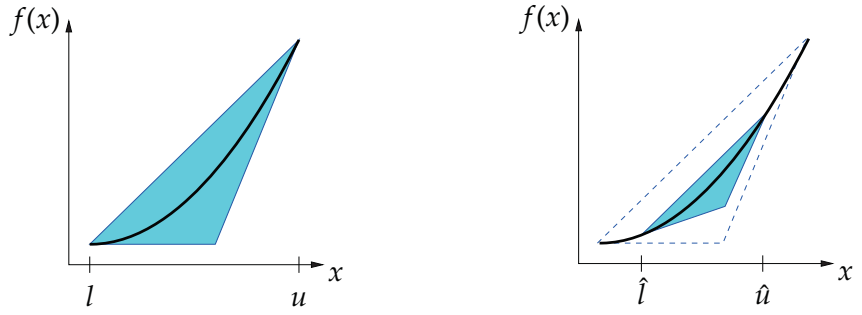
In this chapter we present a bound tightening technique for material balance equations in the framework of hybrid distillation/melt-crystallization processes that reduces the domains significantly and thus accelerates the computations. Our approach leads to *boundary intervals* for the composition variables $x_{i,l}$ 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