Relaxation Refinement for Mixed-Integer Nonlinear Programs with Applications in Engineering

Dissertation
zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

Der Fakultät für Mathematik der
Technischen Universität Dortmund
vorgelegt von

Nick Mertens

am 06.09.2019

Dissertation

Relaxation Refinement for Mixed-Integer Nonlinear Programs with Applications in Engineering

Fakultät für Mathematik
Technische Universität Dortmund

Erstgutachter: Prof. Dr. Christoph Buchheim

Zweitgutachter: Prof. Dr. Frauke Liers

Tag der mündlichen Prüfung: 15.11.2019

Acknowledgement

First of all, I would like to thank my supervisors Dennis Michaels and Christoph Buchheim. They gave me the opportunity to work on this thesis and were always approachable for my problems and questions. Their guidance regarding mathematical issues and academic life in general is much appreciated.

Parts of this thesis are the result of joint works and projects. I would like to thank Christian Kunde, Maximilian Merkert and Frauke Liers for their support and the productive collaborations. In particular, I would like to thank Frauke Liers for her involvement in the examining process. I also want to thank the Deutsche Forschungsgemeinschaft (DFG) for their financial support through most parts of my work.

I am grateful for all former and present colleagues at TU Dortmund for creating a pleasant work environment. Special thanks goes to Felix Hommelsheim for proof-reading parts of this thesis.

Abstract

Solution strategies for Mixed-Integer Nonlinear Programs (MINLPs) often rely on a convex relaxation of the feasible set. This relaxation is used to derive lower bounds and to evaluate the quality of local solutions. In this thesis, we discuss different approaches of constructing and improving suitable relaxations. We further analyze these relaxations with respect to tightness and quality of the resulting lower bounds. This is done for general MINLPs as well as for specific problems arising from certain real world applications.

We develop a cutting plane method for the convex hull of the feasible set of relatively general MINLPs. It is based on simultaneous considerations of the involved constraints and on solving a convex optimization problem. This underlying separation problem is non-differentiable and requires the convex envelope of linear combinations of the constraint functions. We analyze its structure and smoothness in detail, and discuss suitable solution approaches. Furthermore, we introduce approximation strategies for the convex envelope and discuss the resulting approximate version of the separation problem. This approximate version leads to weaker results but to a greater applicability.

The proposed cutting plane approach is further applied to constraint sets consisting of bivariate quadratic absolute value functions. We present general analytic tools and concepts, and derive the convex envelope of the considered functions under certain assumptions. This type of functions also emerges from the modeling of gas networks, which allows us to computationally evaluate the impact of our cutting plane approach on a real world application.

Finally, we consider an example of optimal design problems in chemical engineering. For a distillation column model, we introduce a suitable reformulation and prove monotonic behavior of several sequences of relevant variables. Reformulation and monotonicity are used to improve the formulation of the respective feasible set. In particular, we develop a problem specific bound tightening strategy. Our results are computationally evaluated on multiple test instances.

Zusammenfassung

Lösungsstrategien für Gemischt-Ganzzahlige Nichtlineare Programme (MINLPs) basieren häufig auf einer konvexen Relaxierung der zulässigen Menge. Diese Relaxierung wird benutzt um untere Schranken zu ermitteln und um die Qualität lokaler Lösungen zu beurteilen. In dieser Thesis diskutieren wir verschiedene Ansätze um geeignete Relaxierungen zu konstruieren und zu verbessern. Außerdem analysieren wir diese in Hinblick auf Strenge und Qualität der resultierenden unteren Schranken. Dabei betrachten wir sowohl allgemeine MINLPs als auch spezifische Probleme, die sich aus der Anwendung ergeben.

Wir entwickeln ein Schnittebenenverfahren für die konvexe Hülle der zulässigen Menge von relativ allgemeinen MINLPs. Es basiert auf der simultanen Betrachtung von Nebenbedingungen und auf einem konvexen Optimierungsproblem. Dieses Separationsproblem ist nicht-differenzierbar und benötigt die konvexe Einhüllende von Linearkombinationen der Nebenbedingungen. Wir analysieren seine Struktur und Glätte ausführlich und diskutieren passende Lösungsansätze. Außerdem entwickeln wir Approximationen der konvexen Einhüllenden und ein ensprechendes approximatives Separationsproblem. Dieses führt zu schwächeren Resultaten aber zu einer höheren Anwendbarkeit.

Das obige Schnittebenenverfahren wird außerdem auf eine Menge von Nebenbedingungen angewendet, die aus bivariaten quadratischen Absolutwertfunktionen besteht. Wir präsentieren allgemeine analytische Hilfsmittel und Konzepte und bestimmen die konvexe Einhüllende für diese Funktionen unter gewissen Voraussetzungen. Diese Klasse von Funktionen wird auch bei der Modellierung von Gasnetzwerken verwendet, was es uns erlaubt den Einfluss des Schnittebenenverfahrens auf Probleme aus der Anwendung zu untersuchen.

Schließlich betrachten wir noch ein Beispiel eines optimalen Designproblems aus dem Bereich des Chemieingenieurwesens. Für das Modell einer Destillationskolonne bieten wir eine Reformulierung an und beweisen monotones Verhalten von bestimmten Folgen relevanter Variablen. Reformulierung und Monotonie werden benutzt um die Formulierung der zugehörigen zulässigen Menge zu verbessern. Insbesondere entwickeln wir eine problemspezifische Bound-Tightening-Strategie. Unsere Ergebnisse werden an einigen Testinstanzen computergestützt evaluiert.

Contents

Chapter 1

Introduction

Many theoretical questions and relevant applications can be mathematically formulated as optimization problems. The resulting problem classes are usually distinguished by the type of the involved decision variables and constraint functions. This thesis deals with problems consisting of finitely many discrete as well as continuous variables, and continuous constraints that may be nonlinear and nonconvex. These mixed-integer nonlinear programs (MINLPs) form a rather general, and therefore challenging problem class. In addition to the well-known complexity of integer programming, the nonconvexity of the constraints in general leads to multiple locally optimal solutions that are not globally optimal. Solving MINLPs is therefore also denoted as global optimization, and the required solution strategies are quite sophisticated. In this thesis, we mainly focus on handling the difficulties arising from the nonlinear constraint set in MINLPs.

This is done in two parts. In the first part, we describe the most established solution strategy for general MINLPs called spatial Branch and Bound. One important component of this strategy is the generation of lower bounds by considering a convex relaxation of the feasible set. This thesis focuses on analyzing the quality of this relaxation and on improving it by so-called relaxation refinement strategies. For this, we first introduce a proper theoretical foundation and present relevant results from the literature. Next, we extend these results in order to derive a relaxation refinement for relatively general MINLPs that is based on the interaction of multiple constraint functions. We further discuss properties and applicability of the resulting refinement strategy.

In the second part, we investigate two important applications in the broad field of engineering. Namely, we consider gas network operation and a chemical separation process. A common question in this context is the one of optimal design, either in terms of productivity or economic aspects. These questions can be interpreted as optimization problems that require discrete decision variables and nonlinear constraints to be modeled adequately, i.e., MINLPs. Despite recent progress concerning solution strategies and software, many MINLPs arising from real world applications need an unreasonable amount of solution time. Therefore, it is often useful or necessary to apply problem-specific optimization techniques. These techniques are based on the usual components of the spatial Branch and Bound, but exploit properties of the given application or problem class to be more effective. They are restricted to special types of MINLPs, but are often able to significantly reduce the solution time. Motivated by this, we analyze the structure and properties of the two applications, and use this knowledge to design problem specific relaxation refinement strategies for both.

This thesis is structured as follows. Chapter 2 formally introduces the considered MINLP problem class and the spatial Branch and Bound solution strategy. This includes the generation of upper and lower bounds by heuristics and relaxations, common branching strategies, and the main requirements for convergence. We further discuss relaxation refinement strategies, as they represent the common topic of all following chapters.

In Chapter 3, we introduce the convex envelope and discuss the "standard" way of constructing a convex relaxation of the feasible set. We illustrate that this construction leaves room for improvement. We present a result from the literature that characterizes the best possible convex relaxation. It is based on a simultaneous consideration of the involved constraints. We further exploit this characterization in order to derive a cutting plane method for relatively general MINLPs. The requirements needed to apply this method are high, as it uses the convex envelope of all linear combinations of the constraint functions. Therefore, we also introduce an approximate version that leads to weaker results but to a greater applicability.

Chapter 4 deals with problems arising from gas network operation. We consider a single junction in such a network. The resulting constraints are given as quadratic absolute value functions. We present some general analytic tools and concepts, and derive the convex envelope of the considered functions under certain assumptions. This allows us to apply the cutting plane method from Chapter 3 to the feasible set of gas networks. We exemplarily evaluate the computational impact on two small test instances.

In Chapter 5, we develop a problem specific relaxation refinement strategy for optimal design problems of distillation columns. For this, we first present a detailed model of the considered distillation process and introduce a reformulation. For this reformulation, we prove monotonic behavior of several sequences of relevant variables. We further develop a bound tightening strategy for the considered problem class based on this monotonic behavior. The influence of the presented techniques is evaluated on multiple artificial test instances.

A conclusion is given in Chapter 6. We briefly discuss the connection between the presented results and put them into relation to the state of research.

Several parts of this thesis are the result of joint works. Chapter 4 is based on collaboration with Frauke Liers, Alexander Martin, Maximilian Merkert and Dennis Michaels. Chapter 5 is based on collaboration with Achim Kienle, Christian Kunde and Dennis Michaels. However, the presentation is focused on the results provided by the author. Additional information is given in the respective chapters.

Part I

Mixed-Integer Nonlinear Optimization

Chapter 2

Branch and Bound for MINLPs

We consider an optimization problem with continuous as well as discrete decision variables and a nonlinear constraint set. Such a problem is called a *Mixed-Integer Nonlinear Program* (MINLP) and can be formulated as

Problem 2.1.

$$\begin{aligned} \text{min} \quad & f(x) := c^\top x \\ \text{s.t.} \quad & x \in X \\ & X := \big\{x \in [l, u] \mid g(x) \le 0,\ x_i \in \mathbb{Z}\ \forall\ i \in J\big\}. \end{aligned}$$

We call X *feasible set*, $f : \mathbb{R}^n \to \mathbb{R}$ *objective function* and the entries of $g : \mathbb{R}^n \to \mathbb{R}^m$ *constraints.* Note that the constraints are allowed to be nonlinear and in particular nonconvex. *Integrality constraints* are imposed on a subset of variables by the index set $J \subseteq \{1, \ldots, n\}$. Lower and upper bounds on the variables are given by $l, u \in \big(\mathbb{R} \cup \{\pm\infty\}\big)^n$. We assume that the objective function is linear, as potential nonlinearities can be moved to the constraint set. Common further requirements for practical purposes are real valued bounds l, u and a certain degree of smoothness of the constraints.

Problems that can be modeled as MINLPs arise from many different applications like chemical process design, network operation, and engineering in general (Grossmann et al. [1999]; Martin et al. [2006]). See also [Burer and Letchford, 2012] and [Belotti et al., 2013] for an overview on applications. It is therefore desirable to be able to determine an exact solution, or at least a reliable approximation of Problem 2.1. However, discrete decision variables as well as nonconvex constraint functions on their own make general MINLPs

NP-hard in theory (Kannan and Monma [1978]). Furthermore, the combination makes MINLPs also particularly difficult to solve from a practical point of view.

In this chapter we review a solution strategy for MINLPs called *spatial Branch and Bound*. It can be applied to quite general problem classes and is used in many state-of-the-art software packages like SCIP (Gleixner et al. [2017]), ANTIGONE (Misener and Floudas [2014]) and BARON (Tawarmalani and Sahinidis [2005]). See also [Bussieck and Vigerske, 2011] for a summary of available solvers.

The presented information in this chapter is mostly known for several years and already gathered in various publications. For a very datailed introduction we refer to [Locatelli and Schoen, 2013; Vigerske, 2012; Belotti et al., 2013]. Further publications are available, for example with focus on quadratic problems (Burer and Letchford [2012]), derivative free optimization (Boukouvala et al. [2016]) and the MINLP solver SCIP (Vigerske and Gleixner [2018]).

The remainder of this chapter is structured as follows. In Section 2.1 we highlight some important components of the spatial Branch and Bound solution strategy. This includes the generation of upper and lower bounds by heuristics and relaxations, common branching strategies, and the main requirements for convergence. Section 2.2 deals with relaxation refinement strategies. Therein, we describe methods used to derive additional constraints on the feasible set in order to speed up the solution process of the spatial Branch and Bound.

2.1 General Solution Strategy

The general spatial Branch and Bound solution strategy is based on the following approach. The feasible set of the main problem is successively divided, resulting in multiple subproblems with respective smaller feasible sets. This procedure is called branching, as it can be described with a tree structure consisting of branches of subproblems. We call this structure *Branch and Bound tree.*

For each subproblem, we derive upper and lower bounds on the optimal objective value. Upper bounds are given by any feasible solution and can be computed by heuristics (see Section 2.1.1). Lower bounds are derived by generating and solving a convex relaxation of the respective subproblem (see Section 2.1.2). The convexity of the relaxation ensures that we are able to find its global optimum. The resulting lower bound can be used to evaluate the quality of the heuristic solution. If lower and upper bounds coincide, or are sufficiently close, the heuristic solution is considered as (approximately) optimal for this subproblem. Otherwise another branching step is applied (see Section 2.1.3). It is expected that the bounds are tighter for smaller feasible sets, eventually leading to convergence of this procedure under several assumptions (see Section 2.1.4).

In order to reduce the size of the Branch and Bound tree, it is also important to close branches if they can not contain any additional information for the original problem. This is obviously the case, when a subproblem is found to be infeasible or when upper and lower bound are sufficiently close. Furthermore, when the lower bound of a subproblem is larger than the upper bound of any other subproblem in the tree, it does also not contain an optimal solution and can be deleted. The latter procedure is called *pruning*.

2.1.1 Heuristics

Heuristics can be used as an alternative to an exact solution strategy for Problem 2.1. This is the case if we are not interested in the optimal solution, but only in finding a (good) feasible solution with relatively small computational effort. In the context of this thesis however, it is important to highlight the benefits of good heuristics for the convergence speed of the spatial Branch and Bound.

We already mentioned that every feasible solution gives an upper bound on the respective subproblem in the Branch and Bound tree. However, as subproblems are generated by restricting the feasible set, these bounds are also valid for the original main problem. Usually we are satisfied with approximately optimal solutions, so a good upper bound obviously speeds up the solution process. Furthermore, it helps to prune parts of the Branch and Bound tree without optimal solutions.

In the following, we briefly discuss three different categories of heuristics. Search heuristics are based on the usage of NLP solvers for a continuous relaxation of Problem 2.1 and MILP solvers for an integer linear relaxation. Both relaxations can be solved significantly faster than the original problem. This is either done independently, or different NLP and MILP relaxations are solved iteratively. Integer variables in the NLP run can then be rounded or fixed to prior MILP solutions, while the MILP problem is aimed to find points close to the last NLP solution.

Other strategies aim to obtain a simpler version of the problem by reducing the search space. Undercover heuristics, for example, fix certain variables to reduce the complexity of the constraints. In some cases, it suffices to fix a small amount of variables in order to derive a linear problem that can be handled by LP solvers. In a neighborhood search, the search space is restricted to a neighborhood of a given point. If this point is nearly feasible or close to optimal, we can assume that a feasible or good solution can be found.

Diving heuristics explore the Branch and Bound tree in a depth-first search. Thereby, we quickly reach subproblems with a smaller feasible set, and increase the likelihood of finding a feasible solution with other heuristics or the standard relaxation.

2.1.2 Lower Bounds

A lower bound for Problem 2.1 can be generated by considering a convex relaxation, given as

Problem 2.2.

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \bar{X} \end{aligned}$$

with $\bar{X} \supseteq X$ convex. Note again that the objective function f is already assumed to be linear. The nonconvexity in the feasible set X however, is due to integer decision variables as well as nonconvex constraint functions. The first restriction can be relaxed by dropping the integrality constraints, which is a common strategy in integer programming. In order to handle the nonconvex constraints, we make use of the concept of convex underestimators.

A convex underestimator h^{lo} of h is a convex function with $h^{\text{lo}}(x) \leq h(x)$ on the underlying domain $[l, u]$. A convex relaxed feasible set to Problem 2.1

is then given by

$$\bar{X} := \left\{x \in [l, u] \mid g^{\mathrm{lo}}(x) \leq 0\right\}$$

with $g^{\mathrm{lo}} := (g_1^{\mathrm{lo}}, \dots g_m^{\mathrm{lo}})^\top$ and convex underestimators g_i^{lo} of g_i on D for all $i = 1, \dots, m$. By definition, we have $X \subseteq \bar{X}$ and the optimal value of Problem 2.2 gives a lower bound on the optimal value of Problem 2.1. Furthermore, Problem 2.2 is convex and can be solved to global optimality by common strategies like interior point methods.

Every subproblem in the Branch and Bound tree is of the same form as Problem 2.1, so we can derive lower bounds for every one of them by following the methodology above. In contrast to upper bounds, these lower bounds are only valid for the respective subproblem and further subproblems in this specific branch. They are in particular not valid for the original problem or the entire tree.

An important step in the presented strategy is the construction of convex underestimators. They have a significant influence on the quality of the resulting lower bound. From this point of view, the "best" convex underestimator is called the convex envelope. It is defined as the point-wise supremum over all convex underestimators, which is again a convex underestimator. In general, the convex envelope is hard to determine for arbitrary constraint functions. Therefore, a broad field of research is devoted to deriving the convex envelope for specific classes of functions (e.g., Sherali and Alameddine [1990]; Rikun [1997]; Tawarmalani and Sahinidis [2001]; Meyer and Floudas [2005]; Jach et al. [2008]; Locatelli and Schoen [2014]). See also [Kleibohm, 1967] and [Falk, 1969] for early work and [Boukouvala et al., 2016] for a list of publications on this subject. In Chapter 3, we go into more detail on the construction of the convex envelope. In Chapter 4, we present some tools that are helpful in deriving it for specific classes of functions.

However, a representation of the convex envelope, or even good convex underestimators are in general not given for every constraint function. But from a computational point of view, we need at least a strategy to construct convex underestimators in order to make use of the spatial Branch and Bound. One of the most popular strategies is based on factorization. It is presented in [McCormick, 1976] and implemented in many state-of-the art software packages. Every function is successively divided into smaller parts until a set of basis functions is reached, for which the convex envelope, or a good convex underestimator is available. A suitable composition of these underestimated basis functions does result in a convex underestimator of the original constraint.

Note that the choice of the convex underestimators also influences the computational effort needed to solve the respective relaxed subproblem. Therefore, it is not always beneficial to use the convex envelopes as underestimators, as they tend to have a complicated representation. Instead, convex underestimators with easy representations, like piecewise linear functions are often used during the solution process, although they produce worse lower bounds.

2.1.3 Branching

Branching is the procedure of dividing the feasible set of a problem into several, usually two, smaller sets. The motivation is that the respective lower bounds of the smaller problems are in general tighter.

A standard approach is to divide the feasible set by choosing one variable and splitting its interval into two parts. This strategy originates from integer linear programming. If the solution of the linear relaxation of an ILP is infeasible, it consists of at least one variable i with non-integral value. Let the value of this variable be $x_i \in [l_i, u_i]$. We then design two subproblems with the intervals of variable i given by $\big[l_i, \lfloor x_i \rfloor\big]$ and $\big[\lceil x_i \rceil, u_i\big]$, respectively. As the current relaxed solution is feasible for neither of the subproblems, we expect progress in the next step of the Branch and Bound algorithm.

In the MINLP setting, we also need to branch on continuous variables. This is called *spatial branching* and is done in a similar way. If the solution of a relaxed MINLP is not feasible besides all integral variables meeting their restriction, there have to be some nonlinear constraints producing this infeasibility. Hence, it is reasonable to branch on any variable that appears in one of those constraints. In contrast to the integer case, we can not reduce the overall search space by exploiting integrality constraints. Instead, the new intervals are given by $[l_i, x_i]$ and $[x_i, u_i]$. However, as convex underestimators are usually tight at the boundary of the underlying domain, it is still likely that the relaxation is improved in the next step. Note further that, if the value x_i is close to the boundary, the branching point is usually moved towards the middle of the interval in order to generate a more balanced Branch and Bound tree.

As the choice of the branching variable is in general not unique and has a huge impact on the solution process, there are several different strategies designed to support this decision. Branching primarily aims on improving the lower bounds of the problem in order to speed up the solution process. Hence, it makes sense to choose the variable that results in the greatest improvement of the lower bound as the branching variable. This improvement can either be computed by explicitly solving the relaxation of the resulting subproblems for different choices of branching variables (strong branching), or it can be estimated using data from prior branching steps (pseudocost branching). As the first strategy is computationally expensive and the second one is not reliable in early stages, it is common to combine both approaches (reliability branching).

2.1.4 Convergence

We briefly discuss the convergence of the spatial Branch and Bound algorithm. For this, consider a setting where we are not satisfied with an approximate solution of Problem 2.1. Instead, the algorithm only terminates if lower bound and upper bound coincide, or if infeasibility of the original problem is detected. Otherwise the algorithm generates an infinite sequence of subproblems. We say that the algorithm converges when it either terminates with one of the two results above, or when the lower bounds of the infinite sequence of subproblems converge towards the optimal solution of Problem 2.1. In the following, we cite three important conditions that are commonly demanded to ensure convergence. See [Locatelli and Schoen, 2013, Chap. 5] for a detailed analysis.

The first condition is a *bound improving node selection*. It demands that, after branching, the open subproblem with the smallest lower bound is considered next. The successive number of times where this rule is not applied always has to be finite.

The second condition is the *exactness in the limit* of the used convex underestimators. This means, that the difference between a constraint function g_i and its convex underestimator g_i^{lo} tends to zero if the diameter of the underlying set tends to zero.

The last one is the *exhaustiveness property.* When the algorithm does not terminate after a finite number of iterations, the branching strategy produces infinite sequences of subsets. These subsets need to converge towards a single point, which means that the respective diameters tend to zero.

2.2 Relaxation Refinement

As derived above, one of the most important components of the spatial Branch and Bound solution strategy is the generation of good lower bounds for the considered subproblems. They are essential for the convergence speed, as a globally optimal solution is usually only verified if the lower bound is equal (or very close) to the actual objective value. Furthermore, having good lower bounds at hand allows for more efficient pruning of the Branch and Bound tree.

Problem 2.2 gives the best lower bound for Problem 2.1 if the relaxed feasible set $\bar{X}$ is chosen to be the convex hull of X. This holds, as the convex hull is the inclusion-wise smallest convex superset of X. In fact, the optimal objective values of Problem 2.1 and 2.2 are equal in this case, as the objective function is linear.

However, following the outlined strategy in Section 2.1.2, the relaxed feasible set is in general larger than the convex hull of the feasible set, i.e., $\mathrm{conv}(X) \subsetneq \bar{X}$. There are three main reasons for this. First of all, dropping the integrality constraints does not lead to the convex hull of the integral points. This behavior is well known in integer programming and the reason for its complexity. Second, the convex underestimators are usually not as tight as possible due to the fact that the convex envelopes are not available or undesirable. At last, even if the convex envelopes are used and no integer variables occur, there is in general still a discrepancy between $\mathrm{conv}(X)$ and $\bar{X}$. This is due to the fact that every constraint is considered separately and that the interaction of different constraints is mostly ignored. The latter point is discussed in detail in Chapter 3.

In order to compensate for this, it has become a common strategy to add linear constraints to the relaxed Problem 2.2. These constraints aim to reduce the size of the relaxed feasible set $\bar{X}$, while being valid for the original feasible set X and therefore also for $\text{conv}(X)$. We call this quite general approach *relaxation refinement* and we will highlight two special types of refinement in the following. *Bound tightening* explicitly reduces the ranges of variables, while *cutting planes* are designed to separate given infeasible points from the relaxation.

2.2.1 Bound Tightening

Bound tightening, also known as domain reduction or range reduction, is a common strategy to reduce the initial domain of the problem variables without cutting off the optimal solutions. As tighter variable bounds can be interpreted as additional linear inequalities, this obviously helps in restricting the feasible set of the original problem and its relaxations. Additionally, the explicit bound reduction improves the relaxation even further, as convex underestimators are usually tighter on smaller underlying domains.

In the literature, it is mainly distinguished between two basic types of domain reduction. Feasibility based bound tightening (FBBT) cuts off non-feasible solutions using the constraints of the underlying problem. Standard methods are often based on interval arithmetic (e.g., see Hansen et al. [1991]; Ratschek and Rokne [1995]) and the description of nonlinearities using expression trees (e.g., see Schichl and Neumaier [2005]). Bounds on the variables can be propagated onto the nonlinear expressions via forward propagation. Also, the other way around, tighter bounds on the variables can be computed using the bounds on the nonlinearities (backward propagation). This procedure can be iterated until no further strengthening of the bounds is achieved. In Chapter 5, we develop a specific bound tightening strategy for problems arising from chemical process design.

Optimization based bound tightening (OBBT) applies optimization techniques in order to derive tighter variable bounds. The key idea is to consecutively minimize and maximize every variable on the feasible set. This is in general as hard as finding the optimal solution of the problem itself, so a common approach is to only use (linear) relaxations of the feasible set. This procedure can also be iterated multiple times in order to further tighten the bounds. OBBT can be more effective than FBBT, but is often much more time-consuming. It is therefore used very rarely or only at the root node of the Branch and Bound tree. See [Gleixner et al., 2016] for more information on OBBT and [Quesada and Grossmann, 1993] for an early application.

2.2.2 Cutting Planes

Cutting planes are a special type of linear inequalities. They are designed to separate given points from the feasible set and are commonly used in the context of optimization.

Let a feasible set X, its convex hull $\operatorname{conv}(X)$ and an arbitrary relaxation $\bar{X} \supsetneq \operatorname{conv}(X)$ be given. Furthermore, we assume to have a point $x \in \bar{X}$ which may be a solution of a relaxed problem. If $x \notin \operatorname{conv}(X)$, a cutting plane for x is a linear inequality that is valid for $\operatorname{conv}(X)$ but not valid for x. As $x \notin \operatorname{conv}(X)$ holds, the hyperplane separation theorem states that such an inequality always exists. If we are able to find such a hyperplane, we add it to the relaxed set $\bar{X}$ and thus make sure that the same solution will not be obtained in any further optimization runs. If the cutting planes are chosen "strong enough", then the solution of the relaxed problem converges iteratively towards a point that lies inside $\operatorname{conv}(X)$. See [Kelley, 1960] for an early publication on cutting planes in the context of integer linear programming. In Chapter 3 and 4 we develop and apply a cutting plane approach for MINLPs.

As mentioned above, the objective values of Problem 2.1 and 2.2 are the same for $\bar{X} = \operatorname{conv}(X)$. Hence, well designed cutting planes can be used as an alternative approach to the spatial Branch and Bound in order to converge towards the optimal objective value. More commonly, cutting planes are integrated into the Branch and Bound framework as an additional step prior to the branching. This helps in deriving tighter lower bounds for the considered subproblems and speeds up the solution process. The combination is also called *Branch and Cut*.

Chapter 3

Individual and Simultaneous Convexification

In Chapter 2 we briefly described the main components of the spatial Branch and Bound solution strategy for MINLPs. Recall that an important step for computing lower bounds is the generation of the relaxed feasible set $\bar{X}$ in the relaxed Problem 2.2. In this chapter we focus on handling the nonlinearities in this relaxation. We assume that there are no integrality constraints involved, or that the continuous relaxation is already provided. In order to derive a good relaxed feasible set, a common strategy is to replace every single nonlinear constraint by its convex envelope, i.e., the "best" convex underestimator (e.g., see Locatelli and Schoen [2013]). We call this "standard" approach *individual convexification*, as it considers every single constraint function individually.

However, the convex hull of a feasible set defined by multiple constraint functions is, in general, not completely described by the convex envelope of every single constraint. As a consequence of this observation, the individual convexification of the feasible set can be significantly tightened by considering the interaction between multiple constraint functions. This interaction was already studied in several publications. In this thesis, we follow the nomination in [Tawarmalani, 2010] and refer to the convex hull of a set given by multiple constraints as *simultaneous convexification.* The Reformulation-Linearization Technique (RLT, Sherali and Alameddine [1992]) can be interpreted as an early result in this context. Suitable functions are multiplied in order to derive additional constraints on the relaxation. A combination of the RLT constraints and semidefiniteness is further integrated in [Anstreicher and Burer, 2010] to derive the simultaneous convexification of a set of bivariate quadratic functions. See for example also [Burer and Ye, 2019] for exact semidefinite relaxations of quadratic problems or [Belotti et al., 2010] for the bilinear case.

A more general characterization of the simultaneous convexification is given in [Ballerstein, 2013]. The convex hull of a feasible set, defined as the graph of a vector-valued function, can be described by the convex envelopes of all linear combinations of its components. The author already exploits this result to improve the relaxation of feasible sets given by multiple univariate convex functions. He further identifies linear combinations that are not required for the characterization.

In this chapter, we make use of the result above to derive a refinement of the "standard" relaxation using simultaneous convexification. We present a strategy to include the refinement into an algorithmic framework by a cutting plane method. As a result, we are able to separate from the convex hull of the feasible set of relatively general MINLPs by solving a convex optimization problem. The problem is not differentiable, so that subgradient methods are required for the solution process (e.g., see Lemaréchal [1989]; Mäkelä [2002]; Shor [2012]). Furthermore, the separation problem relies on an algorithmically utilizable representation of the convex envelope of linear combinations of the constraint functions. As the convex envelopes are usually not given for general constraints, we also discuss possible substitutions. Under some assumption, we still derive a necessary condition for the separation strategy to work.

Other publications also focus on cutting planes and supporting hyperplanes for convex envelopes (e.g., see Tawarmalani et al. [2013]; Locatelli and Schoen [2014]). In contrast to those publications, we do not aim to improve the description of the convex envelope of a single constraint function, but of the convex hull of the feasible set given by multiple constraints. The cutting planes in our case are therefore in a higher dimensional space and provide more information. However, the insight from previous work on the general structure of convex envelopes is still very helpful.

The remainder of this chapter is structured as follows. In Section 3.1, we briefly discuss the convex envelope and illustrate that the individual convexification is not sufficient to describe the convex hull of the feasible set. In

Section 3.2, we present a constraint setting for which a representation of the convex hull of the feasible set is given. This result from the literature is used to derive a separation problem based on the convex envelope of linear combinations of the constraint functions. We analyze the smoothness of the separation problem and further present a method to generate cutting planes based on its solution. In Section 3.3, we introduce estimations of the convex envelope that may serve as substitutions in the proposed separation problem. We discuss requirements that still ensure a necessary condition for the separation strategy to work, and present methods for the construction of suitable estimations.

Most fundamental arguments used in the analysis in this chapter are given in more detail in [Rockafellar, 2015]. The author's contribution is mainly presented in Sections 3.2.2 – 3.3.

3.1 Individual Convexification

A common strategy for deriving a convex superset of the feasible set of Problem 2.1 is to replace every single nonlinear constraint by its convex envelope. We call this approach individual convexification. In order to analyze the resulting relaxation quality, we first discuss the convex envelope in more detail.

3.1.1 The Convex Envelope

We make use of the following notation and properties of the convex envelope. See [Locatelli and Schoen, 2013, Chap. 4] for an extensive introduction to this topic.

Definition 3.1. Let $D \subseteq \mathbb{R}^n$ convex and $g : D \to \mathbb{R}$ continuous.

- A convex function $g^{\mathrm{lo}} : D \to \mathbb{R}$ with $g^{\mathrm{lo}}(x) \leq g(x)$ for all $x \in D$ is called a convex underestimator of g on D.
- The convex envelope of g on D is defined by

$$\mathrm{vex}_D[g](\bar{x}) := \sup\big\{h(\bar{x}) \mid h(x) \leq g(x)\ \forall\ x \in D,\ h \text{ convex}\big\}.$$

The value of the convex envelope at a certain point $\bar{x} \in D$ can be determined by the following nonconvex optimization problem.

$$\begin{aligned} \operatorname{vex}_D[g](\bar{x}) = \min & \sum_{i=1}^{n+1} \lambda_i \cdot g(x^i) \\ \text{s.t.} & \sum_{i=1}^{n+1} \lambda_i \, x^i = \bar{x} \\ & \sum_{i=1}^{n+1} \lambda_i = 1 \\ & \lambda_i \geq 0, \quad x^i \in D, \quad i = 1, \dots, n+1. \end{aligned} \tag{3.1}$$

Note that the number of points in (3.1) is bounded by $n+1$ as a consequence of Caratheodory's Theorem.

The restriction $x^i \in D$ in (3.1) is quite general. We can often strengthen the problem formulation significantly by choosing another set $G \subseteq D$ instead of D. For example, if G is a finite set, (3.1) reduces to a linear problem. This holds as x^i can be treated as a parameter instead of a variable. We discuss some valid choices that result in an equivalent problem in the following. In general, the set G can be chosen as a proper subset of D using the concept of generating sets (e.g., see Tawarmalani and Sahinidis [2002]).

Definition 3.2.

- Let $D \subseteq \mathbb{R}^n$. We denote the interior of D by

  $$\operatorname{int}(D) := \big\{x \in D \mid \exists\, U \subseteq D \text{ open with } x \in U\big\},$$

  and the boundary of D by

  $$\operatorname{bd}(D) := D \setminus \operatorname{int}(D).$$

- Let $D \subseteq \mathbb{R}^n$ and $g : D \to \mathbb{R}$. We denote the epigraph of g on D by

  $$\operatorname{epi}(g, D) := \big\{(x, z) \in \mathbb{R}^{n+1} \mid x \in D,\ z \geq g(x)\big\}.$$

- Let $D \subseteq \mathbb{R}^n$ convex. We denote the set of all extreme points of D by

  $$\operatorname{extr}(D) := \big\{x \in D \mid D \setminus \{x\} \text{ is convex}\big\}.$$

- For a continuous function $g : D \to \mathbb{R}$ on a compact convex domain $D \subseteq \mathbb{R}^n$, we denote the generating set of g on D by
$$\mathfrak{G}[g, D] := \Big\{ x \in D \mid \big(x, g(x)\big) \in \operatorname{extr}\Big(\operatorname{conv}\big(\operatorname{epi}(g, D)\big)\Big) \Big\}.$$

By definition of the generating set, we equivalently formulate (3.1) into

$$\begin{aligned} \operatorname{vex}_D[g](\bar{x}) = \min & \sum_{i=1}^{n+1} \lambda_i \cdot g(x^i) \\ \text{s.t.} & \sum_{i=1}^{n+1} \lambda_i \, x^i = \bar{x} \\ & \sum_{i=1}^{n+1} \lambda_i = 1 \\ & \lambda_i \geq 0, \quad x^i \in \mathfrak{G}[g, D], \quad i = 1, \dots, n+1. \end{aligned} \tag{3.2}$$

Next, we provide a necessary condition that allows us to exclude points from the generating set. For this, we require the concept of concave and convex directions.

Definition 3.3. Let a continuous function $g : \mathbb{R}^n \to \mathbb{R}$ be given.

- The set of concave directions of g at $\bar{x} \in \mathbb{R}^n$ is given by
$$\begin{aligned} \delta[g, \bar{x}] := \big\{ d \in \mathbb{R}^n \mid \exists\, \varepsilon > 0 : h_{g,\bar{x},d}(\lambda) &:= g(\bar{x} + \lambda d) \\ & \text{is strictly concave on } [-\varepsilon, \varepsilon] \big\}. \end{aligned}$$

- The set of convex directions of g at $\bar{x} \in \mathbb{R}^n$ is given by
$$\begin{aligned} \xi[g, \bar{x}] := \big\{ d \in \mathbb{R}^n \mid \exists\, \varepsilon > 0 : h_{g,\bar{x},d}(\lambda) &:= g(\bar{x} + \lambda d) \\ & \text{is strictly convex on } [-\varepsilon, \varepsilon] \big\}. \end{aligned}$$

In the case of g being twice continuously differentiable at $\bar{x}$, we obtain

$$\begin{aligned} \delta[g, \bar{x}] &= \big\{ d \in \mathbb{R}^n \mid d^\top H_g(\bar{x}) d < 0 \big\} \\ \text{and} \quad \xi[g, \bar{x}] &= \big\{ d \in \mathbb{R}^n \mid d^\top H_g(\bar{x}) d > 0 \big\}, \end{aligned}$$

where $H_g(\bar{x})$ denotes the Hessian Matrix of g at $\bar{x}$.

A necessary condition for an interior point $x \in \operatorname{int}(D)$ being an element of the generating set is that the function g must be strictly locally convex at x. This leads to the following observation

Observation 3.4. [Tawarmalani and Sahinidis, 2002, Cor. 5] *Let $D \subseteq \mathbb{R}^n$ be convex and $g : D \to \mathbb{R}$ continuous. Then, for every $x \in \operatorname{int}(D)$ with $\delta[g, x] \neq \emptyset$, we have $x \notin \mathfrak{G}[g, D]$.*

In order to derive the convex envelope for a specific function, it is often not useful to restrict the feasible set to $\mathfrak{G}[g, D]$ as done in (3.2). Instead, we may choose a superset $G \supseteq \mathfrak{G}[g, D]$ with $G \subseteq D$ and consider the respective problem

$$\begin{aligned} \operatorname{vex}_D[g](\bar{x}) = \min & \sum_{i=1}^{n+1} \lambda_i \cdot g(x^i) \\ \text{s.t.} & \sum_{i=1}^{n+1} \lambda_i \, x^i = \bar{x} \\ & \sum_{i=1}^{n+1} \lambda_i = 1 \\ & \lambda_i \geq 0, \quad x^i \in G, \quad i = 1, \dots, n+1. \end{aligned} \tag{3.3}$$

In certain cases, this allows us to find an optimal solution $\lambda^\star$ and $\{x^{1,\star}, \dots, x^{n+1,\star}\}$ of (3.3) with a reduced number of non-zero components of the vector $\lambda^\star \in \mathbb{R}^{n+1}$ compared to an optimal solution of (3.2). To indicate non-zero components, we define the support of a vector $\lambda^\star$ as

$$I(\lambda^\star) := \big\{ i \in \{1, \dots, n+1\} \mid \lambda_i^\star > 0 \big\}.$$

Definition 3.5. For $D \subseteq \mathbb{R}^n$, $g : D \to \mathbb{R}$, $\bar{x} \in D$ and some $G \subseteq D$ with $\mathfrak{G}[g, D] \subseteq G$, let $\{\lambda^\star; x^{1,\star}, \dots, x^{n+1,\star}\}$ denote an optimal solution to (3.3) such that the cardinality $|I(\lambda^\star)|$ is minimal. Then we call

$$\mathcal{S}_{g,G}(\bar{x}) := \operatorname{conv}\big(\{x^{i,\star} \mid i \in I(\lambda^\star)\}\big)$$

a minimizing simplex for $\bar{x}$ w.r.t. g and G. If $|I(\lambda^\star)| \leq 2$, then we also use the term minimizing segment for $\mathcal{S}_{g,G}(\bar{x})$.

Remark 3.6. Note that the condition $|I(\lambda^\star)| \leq 2$ results in possible minimizing segments consisting of only one point. This way, points from the generating set formally also have a minimizing segment.

We consider the following small example to show the difference in the number of non-zero components in (3.2) and (3.3) for $G \supsetneq \mathfrak{G}[g, D]$.

Example 3.7. Let $g(x) = -x_1x_2$ and $D = \operatorname{conv}\left(\{(0,0),(0,1),(1,2)\}\right)$. Assume that we are interested in deriving the value of the convex envelope of g at $\bar{x} = (\frac{1}{3}, 1)$. As g is concave on D, the generating set is given by

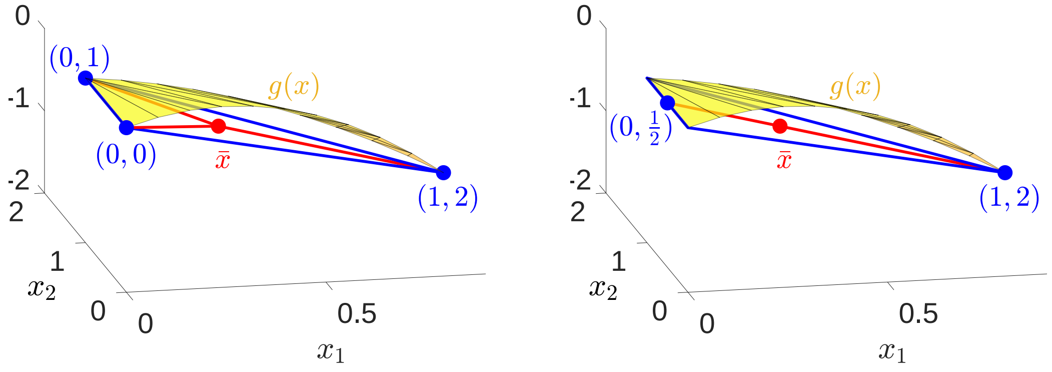
$$\mathfrak{G}[g,D] = \big\{(0,0),(0,1),(1,2)\big\}.$$

The only optimal solution of (3.2) (up to permutations) is therefore $\Big(\lambda^* = \big(\frac{1}{3},\frac{1}{3},\frac{1}{3}\big);\, x^* = \big((0,0),(0,1),(1,2)\big)\Big)$ with three non-zero components in λ^* (see Figure 3.1(a)).

For (3.3) with $G = \operatorname{conv}\Big(\big\{(0,0),(0,1)\big\}\Big) \cup \big\{(1,2)\big\}$, one optimal solution is given by $\Big(\lambda^* = \big(\frac{2}{3},\frac{1}{3},0\big);\, x^* = \big((0,\frac{1}{2}),(1,2),(0,0)\big)\Big)$. The number of non-zero components is two in this case, resulting in a minimizing segment for $\bar{x}$ w.r.t. g and G (see Figure 3.1(b)). It is given by

$$\mathcal{S}_{g,G}\left(\bar{x}\right) := \operatorname{conv}\Big(\big\{(0,\tfrac{1}{2}),(1,2)\big\}\Big).$$

In general, the number of non-zero components can be reduced by this approach if the considered function is linear between two points of the generating set.



(a) Minimizing simplex for $\bar{x}$ w.r.t. $\mathfrak{G}[g,D]$.

(b) Minimizing segment for $\bar{x}$ w.r.t. G.

Figure 3.1: Visualization of Example 3.7.

We can exclude points and pairs of points from being part of minimizing simplices by again using the concept of concave directions.

Observation 3.8. *Let $D \subseteq \mathbb{R}^n$ convex, $g : D \to \mathbb{R}$ continuous and $\bar{x} \in D$. Let $x^i \in \operatorname{extr}\left(\mathcal{S}_{g,D}\left(\bar{x}\right)\right)$ for $i = 1, \dots, m$ with $x^i \neq x^j$ for $i \neq j$.*

- *If $x^i \in \operatorname{int}(D)$ holds for some $i \in \{1, \dots, m\}$, then $\delta[g, x^i] = \emptyset$.*

- *For every pair x^i, x^j, $i, j \in \{1, \dots, m\}$ with $i \neq j$, there exists some $x' \in \operatorname{conv}\left(\{x^i, x^j\}\right)$ with*

$$(x^i - x^j) \in \delta[g, x'].$$

In order to determine the convex envelope of a specific function, it is advantageous to know the dimension of the minimizing simplices beforehand. If there exists, for some $G \subseteq D$ and for every $x \in D$, a minimizing segment w.r.t. g and G, we say that the convex envelope of g (on D) *consists of minimizing segments w.r.t. G*. Note that in this case the convex envelope of g also consists of minimizing segments with respect to every superset $\bar{G} \supseteq G$. These considerations are applied in Chapter 4 to derive the convex envelope of absolute value functions.

However, we are now able to discuss the relaxation quality of the individual convexification using the definition of the convex envelope. This is done in the following subsection.

3.1.2 Relaxation Quality

We consider Problem 2.1 without integrality constraints. The feasible set is given by

$$X = \left\{x \in [l, u] \mid g(x) \leq 0\right\}.$$

We briefly discuss different approaches of constructing a suitable relaxed feasible set $\bar{X} \supseteq X$ for Problem 2.2. The choice of $\bar{X}$ is crucial for the quality of the lower bound attained by Problem 2.2 and has a significant influence on the solution process of the spatial Branch and Bound.

We use the following basic definitions.

Definition 3.9. Let $D \subseteq \mathbb{R}^n$ convex and $g : D \to \mathbb{R}^m$ continuous.

- Let $A, B \subseteq \mathbb{R}^n$. We say A is tighter than B if $A \subseteq B$ holds.
- Let $g^{\mathrm{lo},1}, g^{\mathrm{lo},2} : D \to \mathbb{R}^m$ be two convex underestimators of g. We say $g^{\mathrm{lo},1}$ is tighter than $g^{\mathrm{lo},2}$ if $\mathrm{epi}(g^{\mathrm{lo},1}, D) \subseteq \mathrm{epi}(g^{\mathrm{lo},2}, D)$ holds.

Problem 2.2 gives the best lower bound for Problem 2.1 if $\bar{X}$ is chosen as tight as possible. The tightest possible convex superset of X is $\mathrm{conv}(X)$. As $\mathrm{conv}(X)$ is hard to determine in general, a standard approach is to use $\bar{X} = X^0$ with

$$X^0 := \big\{x \in [l, u] \mid g^{\mathrm{lo}}(x) \leq 0\big\},$$

$g^{\mathrm{lo}} := (g_1^{\mathrm{lo}}, \ldots, g_m^{\mathrm{lo}})^\top$ and convex underestimators g_i^{lo} of g_i for $i = 1, \ldots, m$. It is easy to see that tighter estimators lead to a tighter relaxed feasible set X^0. By definition, the tightest possible estimators are the convex envelopes. Hence we define

$$X^\star := \big\{x \in [l, u] \mid g^\star(x) \leq 0\big\},$$

with $g^\star := \big(\mathrm{vex}_D[g_1], \ldots, \mathrm{vex}_D[g_m]\big)^\top$. We call $X^\star$ individual convexification of X.

In general, we have $X^\star \subsetneqq X^0$ and $\mathrm{conv}(X) \subsetneqq X^\star$. The latter statement is demonstrated in the following example (see also [Ballerstein, 2013]).

Example 3.10. Consider the set

$$\begin{aligned} X &= \big\{x \in [0,1]^3 \mid x_2 = x_1^2, x_3 = x_1^3\big\} \\ &= \big\{x \in [0,1]^3 \mid x_2 \leq x_1^2, -x_2 \leq -x_1^2, x_3 \leq x_1^3, -x_3 \leq -x_1^3\big\} \end{aligned}$$

Both functions $g_1(x) := x^2$ and $g_2(x) := x^3$ are already convex and the convex envelopes of their negatives are given by

$$\mathrm{vex}_D[-g_1](x) = \mathrm{vex}_D[-g_2](x) = -x$$

for $D = [0, 1]$. Therefore we obtain

$$X^* = \big\{x \in [0,1]^3 \mid x_1^2 \leq x_2 \leq x_1, x_1^3 \leq x_3 \leq x_1\big\}.$$

We show that $\operatorname{conv}(X) \subsetneq X^*$ holds. For this, consider maximizing the linear objective function

$$h(x) := x_1 + 2(x_3 - x_2)$$

on $\operatorname{conv}(X)$ and $X^\star$ respectively. As $h(x)$ is linear, we have

$$\max_{x \in X} h(x) = \max_{x \in \operatorname{conv}(X)} h(x).$$

The statement follows by showing

$$\max_{x \in X} h(x) < \max_{x \in X^*} h(x).$$

In fact, $x_2 = x_1^2$ and $x_3 = x_1^3$ hold for $x \in X$. We derive

$$\max_{x \in X} h(x) = \max_{x_1 \in [0,1]} x_1 + 2(x_1^3 - x_1^2) \leq 1.$$

On the other hand, $(0.9, 0.81, 0.9) \in X^*$ holds with $h(0.9, 0.81, 0.9) = 1.08 > 1$.

Example 3.10 indicates that the resulting lower bound obtained by the relaxed Problem 2.2 with $\bar{X} = X^0$, or even $\bar{X} = X^*$ is not the best possible. This observation is the motivation for considering the interaction between different constraints, as it is done in the following section.

3.2 A Separation Method using Simultaneous Convexification

We develop a separation method for MINLPs based on a result on simultaneous convexification. Compared to the individual convexification, it provides additional information by considering the interaction between multiple constraint functions. The resulting separation problem is convex but not continuously differentiable. Furthermore, it relies on an algorithmically utilizable representation of linear combinations of the involved constraint functions. It can be applied whenever the feasible set, or a subset of it, is given as the graph of a vector-valued function.

First, we present a result from the literature and briefly discuss the constraint set needed to make use of it. Afterwards, we develop the separation problem, analyze the objective function, and present a way to derive linear inequalities based on the solution of this problem.

3.2.1 Representation of the Simultaneous Convex Hull

We are interested in deriving the convex hull of the feasible set of MINLPs, as it is an important tool for the construction of tight lower bounds in the Branch and Bound framework. We discussed in Section 3.1, that the individual convexification is in general not sufficient to describe the convex hull of the feasible set. Example 3.10 indicates that this is in particular the case when multiple variables depend on different constraints with the same arguments. Therefore, we consider a feasible set given as the graph of a vector-valued function. The resulting MINLP has the form of

Problem 3.11.

$$\begin{aligned} \min \quad & c^\top(x,z) \\ \text{s.t.} \quad & (x,z) \in X \\ & X := \big\{(x,z) \mid z = g(x),\ x \in D\big\} \end{aligned}$$

with a cost vector $c \in \mathbb{R}^{n+m}$, a compact and convex set $D \subseteq \mathbb{R}^n$ and a continuous function $g : D \to \mathbb{R}^m$. Integrality constraints are omitted, as this chapter focuses on handling the nonlinearities in MINLPs.

However, even without the integrality constraints, not all MINLPs of the form of Problem 2.1 can be formulated this way. Only certain types of dependencies are allowed in Problem 3.11 and bounds on z are only given implicitly by x. This is a relevant restriction for general MINLPs. Nevertheless, the proposed structure is well suited for demonstrating the difference between individual and simultaneous convexification. Furthermore, at least a substructure of the form X is given in almost any MINLP. The developed strategies may therefore still be applied in order to tighten the relaxation of more general feasible sets.

In order to derive a lower bound for Problem 3.11, we consider the relaxed problem given as

Problem 3.12.

$$\begin{aligned} \min \quad & c^\top(x,z) \\ \text{s.t.} \quad & (x,z) \in \bar{X} \end{aligned}$$

with a convex superset $\bar{X} \supseteq X$ (see Chapter 2). The tightest possible choice for $\bar{X}$ is the convex hull of the feasible set

$$Y := \operatorname{conv}(X) = \operatorname{conv}\big(\{(x, g(x)) \mid x \in D\}\big).$$

As the individual convexification is in general not sufficient to describe Y, we make use of a result in [Ballerstein, 2013] instead. The convex hull of a vector of continuous functions on a compact, convex domain can be described using the convex envelopes of all possible linear combinations of its entries.

Proposition 3.13. [Ballerstein, 2013, Cor. 5.25] *Let $D \subseteq \mathbb{R}^n$ be a compact, convex domain and $g : D \to \mathbb{R}^m$ a continuous function with $x \mapsto g(x)$, $g(x) := \big(g_1(x), \ldots, g_m(x)\big)^\top$. Then it is*

$$Y = \bigcap_{\alpha \in \mathbb{R}^m} M_g(\alpha)$$

with

$$M_g(\alpha) := \big\{(x, z) \in \mathbb{R}^{n+m} \mid \alpha^\top z \geq \mathrm{vex}_D[\alpha^\top g](x),\, x \in D\big\}.$$

In the following, we use this representation to derive a convex optimization problem that provides suitable α for separating points from the convex hull of the feasible set of Problem 3.11. Furthermore, we present a strategy to compute cutting planes based on this α.

Our result can be algorithmically exploited by a cutting plane method (see Section 2.2.2). First, the relaxed Problem 3.12 is solved with an arbitrary $\bar{X} \supseteq \mathrm{conv}(X)$. Second, a linear inequality that separates the optimal solution from $\mathrm{conv}(X)$ is added to the description of $\bar{X}$. This way, the feasible set of Problem 3.12 becomes tighter with every iteration.

3.2.2 Separation Problem

Based on Proposition 3.13, we derive an algorithmic framework for the following separation task.

Separation Task 3.14.

`Input:` A compact and convex set $D \subseteq \mathbb{R}^n$, a continuous function $g : D \to \mathbb{R}^m$ and a point $(\bar{x}, \bar{z}) \in D \times \mathbb{R}^m$.

`Task:` Decide whether $(\bar{x}, \bar{z}) \in Y$ and, if not, return a vector $\alpha \in \mathbb{R}^m$ with

$$(\bar{x}, \bar{z}) \notin M_g(\alpha).$$

Let a point $(\bar{x}, \bar{z}) \in D \times \mathbb{R}^m$ be given. According to Proposition 3.13, we have

$$\begin{aligned} (\bar{x}, \bar{z}) \notin Y \quad &\Leftrightarrow \quad \exists\, \alpha \in \mathbb{R}^m : \ (\bar{x}, \bar{z}) \notin M_g(\alpha) \\ &\Leftrightarrow \quad \exists\, \alpha \in \mathbb{R}^m : \ \mathrm{vex}_D[\alpha^\top g](\bar{x}) > \alpha^\top \bar{z}. \end{aligned}$$

Observe that due to scaling, it suffices to consider only linear multipliers $\alpha \in \mathbb{R}^m$ from the unit ball $B^m := \{\alpha \in \mathbb{R}^m \mid ||\alpha||_2 \leq 1\}$. Thus, we can validate whether the given point $(\bar{x}, \bar{z}) \in D \times \mathbb{R}^m$ is contained in Y by solving the Separation Problem given as

Problem 3.15.

$$\min_{\alpha \in B^m} h(\alpha) := \alpha^\top \bar{z} - \mathrm{vex}_D[\alpha^\top g](\bar{x}).$$

Problem 3.15 has the following properties.

Proposition 3.16.

1. *For all $\alpha \in \mathbb{R}^m$, $h(\alpha) < 0$ holds if and only if $(\bar{x}, \bar{z}) \notin M_g(\alpha)$.*
2. *Problem 3.15 is convex and $h : \mathbb{R}^m \to \mathbb{R}$ is continuous on B^m. In particular, there exists an optimal solution to Problem 3.15.*
3. *Let $\alpha^\star$ be an optimal solution to Problem 3.15. Then $h(\alpha^\star) \geq 0$ holds if and only if $(\bar{x}, \bar{z}) \in Y$.*

Proof.

1/3. By construction and Proposition 3.13.

2. The feasible set B^m is compact and convex, and $\alpha^\top \bar{z}$ is linear in α. Moreover, observe that $\mathrm{vex}_D[\alpha^\top g](\bar{x})$ is concave in α. In fact, for arbitrary $\alpha, \beta \in \mathbb{R}^m$ and $\lambda \in [0, 1]$ we obtain

$$\begin{aligned} &\lambda\, \mathrm{vex}_D[\alpha^\top g](\bar{x}) + (1 - \lambda)\, \mathrm{vex}_D[\beta^\top g](\bar{x}) \\ = \ &\mathrm{vex}_D[\lambda\, \alpha^\top g](\bar{x}) + \mathrm{vex}_D[(1 - \lambda)\beta^\top g](\bar{x}) \\ \overset{\text{(i)}}{\leq} \ &\mathrm{vex}_D[\lambda\, \alpha^\top g + (1 - \lambda)\beta^\top g](\bar{x}) \\ = \ &\mathrm{vex}_D[(\lambda\alpha + (1 - \lambda)\beta)^\top g](\bar{x}) \end{aligned}$$

Inequality (i) holds, as $\mathrm{vex}_D[f_1] + \mathrm{vex}_D[f_2]$ is a convex underestimator of $f_1 + f_2$ for arbitrary $f_1, f_2 : D \to \mathbb{R}$.

Thus, the objective function h of Problem 3.15 is convex on any open superset of B^m and therefore also continuous on B^m.

□

Note that, in the case of $(\bar{x}, \bar{z}) \notin Y$, it is not necessary to solve Problem 3.15 to optimality in order to fulfill Separation Task 3.14. In fact, it suffices to find a point $\alpha \in B^m$ with objective value $h(\alpha) < 0$ to derive $(\bar{x}, \bar{z}) \notin Y$.

From a practical point of view, it is important to mention that an efficient solvability of Problem 3.15 heavily relies on the availability of an algorithmically utilizable representation of the convex envelope of $\alpha^\top g$ for every $\alpha \in B^m$. Moreover, the function $\mathrm{vex}_D[\alpha^\top g]$ is in general not continuously differentiable in α, as discussed in the following subsection.

3.2.3 Smoothness of the Objective Function

We analyze the structure and smoothness of the objective function of the proposed Separation Problem 3.15. For this, consider a fixed point $\bar{x} \in D$ and a fixed continuous vector-valued function $g : \mathbb{R}^n \to \mathbb{R}^m$. We neglect the linear part of the objective function and focus on the convex envelope. As the optimization variable is α, we interpret the convex envelope of $\alpha^\top g$ at $\bar{x}$ as a function in α. Based on Section 3.1 and the identity

$$\sum_{i=1}^{n+1} \lambda_i \cdot \alpha^\top g(x^i) = \alpha^\top \sum_{i=1}^{n+1} \lambda_i \cdot g(x^i),$$

the value of this function can be written as

$$\begin{aligned} f(\alpha) := \mathrm{vex}_D[\alpha^\top g](\bar{x}) = \min\ & \alpha^\top \sum_{i=1}^{n+1} \lambda_i \cdot g(x^i) \\ \text{s.t.} & \sum_{i=1}^{n+1} \lambda_i\, x^i = \bar{x} \\ & \sum_{i=1}^{n+1} \lambda_i = 1 \\ & \lambda_i \geq 0, \quad x^i \in D, \quad i = 1, \dots, n+1. \end{aligned} \tag{3.4}$$

Function f can be considered as the minimum of infinitely many linear functions. For this, we interpret the term $\sum_{i=1}^{n+1} \lambda_i \cdot g(x^i)$ as a coefficient and see that the set of coefficients

$$Q := \left\{ \sum_{i=1}^{n+1} \lambda_i \cdot g(x^i) \mid \sum_{i=1}^{n+1} \lambda_i \, x^i = \bar{x}, \ \sum_{i=1}^{n+1} \lambda_i = 1, \right. \\ \left. \lambda_i \geq 0, \ x^i \in D, \ i = 1, \dots, n+1 \right\}$$

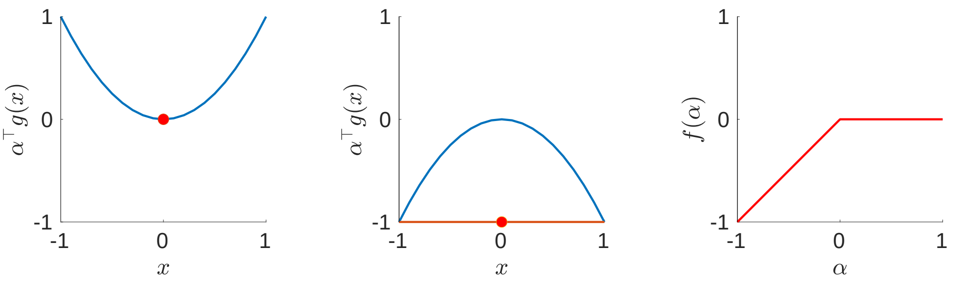
does not depend on α, but on $\bar{x}, g$ and D. Note that Q is compact, as the image of a compact set under a continuous function is compact again. Note further that Q is not convex in general, as g is nonconvex.

Function f is continuous as shown in Proposition 3.16. However, it is in general not continuously differentiable at every point $\alpha \in B^m$. In order to show this statement, consider the following example.

Example 3.17. Let $g(x) = x^2$, $D = [-1, 1]$, $\alpha^\top g(x) = \alpha x^2$ with $\alpha \in B^1$, and $\bar{x} = 0$ be given. For $\alpha \geq 0$, the value of the convex envelope of $\alpha^\top g$ at $\bar{x}$ is equal to zero (see Figure 3.2(a)). For $\alpha < 0$ it can be determined by the minimizing segment $[-1, 1]$ (see Figure 3.2(b)). This leads to the following function for the value of the convex envelope at $\bar{x}$ w.r.t. α (see Figure 3.2(c))

$$f(\alpha) = \text{vex}_D[\alpha^\top g](\bar{x}) = \begin{cases} 0, & \text{if } \alpha \geq 0, \\ -\alpha, & \text{if } \alpha < 0. \end{cases}$$

This function is not continuously differentiable at $\alpha = 0$.



(a) Convex envelope of $\alpha^\top g$ at $\bar{x}$ for $\alpha = 1$.

(b) Convex envelope of $\alpha^\top g$ at $\bar{x}$ for $\alpha = -1$.

(c) Value of the convex envelope at $\bar{x}$ w.r.t. α.

Figure 3.2: Visualization of Example 3.17.

The main reason that f is not differentiable in this case lies in the behavior of the minimizing simplex. For $\alpha < 0$, the minimizing simplex is $[-1,1]$ and for $\alpha \geq 0$, the minimizing simplex is simply $\{0\}$. The extreme points do not change continuously and the resulting simplex is not full dimensional at $\alpha = 0$. This leads to the conjecture that the opposite holds, and that f is differentiable when the minimizing simplex does change continuously and is full dimensional.

We confirm this conjecture by generalizing the setting. For this, we simply consider $\varphi : \mathbb{R}^m \to \mathbb{R}$ as the minimum of infinitely many linear functions, i.e.,

$$\begin{aligned} \varphi(\alpha) := \min\ & c^\top \alpha \\ \text{s.t. }\ & c \in C \end{aligned} \tag{3.5}$$

with a compact (nonconvex) set $C \subseteq \mathbb{R}^m$.

The optimal value of (3.5) exists, so we chose one optimal solution for every $\alpha \in \mathbb{R}^m$ and denote this choice by $L(\alpha)$, i.e.,

$$\begin{aligned} L(\alpha) \in \arg\min\ & c^\top \alpha \\ \text{s.t. }\ & c \in C. \end{aligned}$$

$L(\alpha)$ represents the coefficients of one of the linear functions that generate the minimum of (3.5) at α. We conclude the following results in this setting. Note that the first two statements are already known.

Theorem 3.18. *Let C, φ and L be as defined above.*

1. *φ is concave*

2. *$L(\alpha)$ is a supergradient of φ at α, i.e.,*

$$\varphi(\alpha_0) + L(\alpha_0)^\top (\alpha - \alpha_0) \geq \varphi(\alpha)$$

for all $\alpha_0 \in \mathbb{R}^m$.

3. *If L is continuous at α, then φ is continuously differentiable at α.*

Proof.

1. Let $\alpha, \beta \in \mathbb{R}^m$ and $\lambda \in [0,1]$. It is

$$\begin{aligned}
&\lambda\varphi(\alpha) + (1-\lambda)\varphi(\beta) \\
&= \lambda L(\alpha)^\top \alpha + (1-\lambda)L(\beta)^\top \beta \\
&\overset{\text{(i)}}{\leq} \lambda L\big(\lambda\alpha + (1-\lambda)\beta\big)^\top \alpha + (1-\lambda)L\big(\lambda\alpha + (1-\lambda)\beta\big)^\top \beta \\
&= L\big(\lambda\alpha + (1-\lambda)\beta\big)^\top \big(\lambda\alpha + (1-\lambda)\beta\big) \\
&= \varphi\big(\lambda\alpha + (1-\lambda)\beta\big).
\end{aligned}$$

   The inequality (i) holds, as $L(\alpha)$ is the argmin of (3.5) and $L\big(\lambda\alpha + (1-\lambda)\beta\big) \in C$.

2. Let $\alpha_0 \in \mathbb{R}^m$. With the same argument as above and $L(\alpha_0) \in C$, we obtain

$$\begin{aligned}
\varphi(\alpha_0) + L(\alpha_0)^\top(\alpha - \alpha_0) &= L(\alpha_0)^\top \alpha_0 + L(\alpha_0)^\top(\alpha - \alpha_0) \\
&= L(\alpha_0)^\top \alpha \\
&\geq L(\alpha)^\top \alpha = \varphi(\alpha).
\end{aligned}$$

3. $L(\alpha)$ is a supergradient and therefore the only feasible choice for a possible gradient. Hence, function φ is differentiable at α if

$$\lim_{v \to 0} \frac{|r(v)|}{||v||_2} = 0$$

   holds for $r(v) : \mathbb{R}^m \to \mathbb{R}$ with

$$r(v) := \varphi(\alpha + v) - (\varphi(\alpha) + L(\alpha)^\top v).$$

   It is

$$\begin{aligned}
|r(v)| &= |\varphi(\alpha) + L(\alpha)^\top v - \varphi(\alpha + v)| \\
&\overset{\text{(i)}}{=} \varphi(\alpha) + L(\alpha)^\top v - \varphi(\alpha + v) \\
&= L(\alpha)^\top \alpha + L(\alpha)^\top v - L(\alpha + v)^\top(\alpha + v) \\
&\overset{\text{(ii)}}{\leq} L(\alpha + v)^\top \alpha + L(\alpha)^\top v - L(\alpha + v)^\top(\alpha + v) \\
&= L(\alpha)^\top v - L(\alpha + v)^\top v \\
&= \big(L(\alpha) - L(\alpha + v)\big)^\top v.
\end{aligned}$$

The equation (i) holds, as $L(\alpha)$ is a supergradient of φ at α, and the inequality (ii) holds again with $L(\alpha+v) \in C$. Using Cauchy-Schwartz and the fact that L is continuous at α, we have

$$\begin{aligned}\lim_{v\to 0} \frac{|r(v)|}{||v||_2} &\leq \lim_{v\to 0} \frac{\big(L(\alpha) - L(\alpha+v)\big)^\top v}{||v||_2}\\ &\leq \lim_{v\to 0} \frac{||L(\alpha) - L(\alpha+v)||_2 \cdot ||v||_2}{||v||_2}\\ &= \lim_{v\to 0} |L(\alpha) - L(\alpha+v)||_2 = 0.\end{aligned}$$

Hence, φ is differentiable at α and its gradient is given by $L(\alpha)$. As L is continuous at α, φ is continuously differentiable at α.

□

Next, we transfer this result back to the case of the convex envelope. We choose a solution of (3.4) for every α and denote this choice by $\lambda(\alpha)$ and $x^1(\alpha), \ldots, x^{n+1}(\alpha)$.

Corollary 3.19. *Let $f(\alpha) = \mathrm{vex}_D[\alpha^\top g](\bar{x})$,*

$$K(\alpha) := \begin{pmatrix} \sum_{i=1}^{n+1} \lambda_i(\alpha) \cdot g_1(x^i(\alpha)) \\ \vdots \\ \sum_{i=1}^{n+1} \lambda_i(\alpha) \cdot g_n(x^i(\alpha)) \end{pmatrix}$$

and let $g, x^1(\alpha), \ldots, x^{n+1}(\alpha)$ and $\lambda(\alpha)$ be as defined above. $K(\alpha)$ is a supergradient of f at α. If $K(\alpha)$ is continuous at $\bar{\alpha} \in B^m$, then the function f is differentiable at $\bar{\alpha}$. This holds in particular if the points $x^1(\bar{\alpha}), \ldots, x^{n+1}(\bar{\alpha})$ are affinely independent and $x^i(\alpha)$ is continuous at $\bar{\alpha}$ for all $i = 1, \ldots, n+1$.

Proof. The first two statements follow directly from the definition of (3.4) and Theorem 3.18.

For the last statement, consider $A \in \mathbb{R}^{(n+1)\times(n+1)}$ and $x' \in \mathbb{R}^{n+1}$ given as

$$A(\alpha) := \begin{pmatrix} x^1(\alpha) & \ldots & x^{n+1}(\alpha) \\ 1 & \ldots & 1 \end{pmatrix}, \qquad x' := \begin{pmatrix} \bar{x} \\ 1 \end{pmatrix}.$$

With $x^1(\bar{\alpha}), \ldots, x^{n+1}(\bar{\alpha})$ affinely independent, we derive that $\lambda(\bar{\alpha})$ is the unique solution of

$$A(\bar{\alpha})y = x', \qquad y \in \mathbb{R}^{n+1}.$$

This leads to $\det\big(A(\bar{\alpha})\big) > 0$. As $A(\alpha)$ only depends on the entries of $x^1(\alpha), \dots, x^{n+1}(\alpha)$, and $x^i(\alpha)$ is continuous at $\bar{\alpha}$ for all $i = 1, \dots, n+1$, there is a neighborhood U of $\bar{\alpha}$ with

$$\det\big(A(\alpha)\big) > 0$$

for all $\alpha \in U$. Now we consider a sequence of points $(\alpha^j)_{j\in\mathbb{N}} \subseteq U$ with $\lim_{j\to\infty} \alpha^j = \bar{\alpha}$. With $\det\big(A(\alpha^j)\big) > 0$ we set λ^j to be the unique solution of

$$A(\alpha^j)y = x', \qquad y \in \mathbb{R}^{n+1}$$

for $j \in \mathbb{N}$. The entries of $A(\alpha)$, and therefore also the entries of $A(\alpha)^{-1}$ are continuous at $\bar{\alpha}$. We derive

$$\lim_{j\to\infty} \lambda^j = \lim_{j\to\infty} A(\alpha^j)^{-1}x' = A(\bar{\alpha})^{-1}x' = \bar{\lambda}.$$

We conclude that $\lambda(\alpha)$ and $x^i(\alpha)$ are continuous at $\bar{\alpha}$ for all $i = 1, \dots, n+1$. Function g is continuous, so we derive that $K(\alpha)$ is continuous at $\bar{\alpha}$ and our statement follows. □

Remark 3.20. Note that there are possibly multiple optimal solutions of (3.4) and that $x^1(\alpha), \dots, x^{n+1}(\alpha)$ only denotes one of them. This indicates that there only has to be a unique choice between all possible solutions such that the respective coefficients of this choice behave continuously in the considered point. It may be beneficial to consider different choices of $x^1(\alpha), \dots, x^{n+1}(\alpha)$ in order to show differentiability of $f(\alpha)$. For instance, any permutation of $x^1(\alpha), \dots, x^{n+1}(\alpha)$ is optimal as well, so that a simple lexicographic ordering of the points is already useful.

Note that the specific criterion of affine independence in Corollary 3.19 is in general not necessary for $f(\alpha)$ being differentiable at $\bar{\alpha}$. For this, consider the following example.

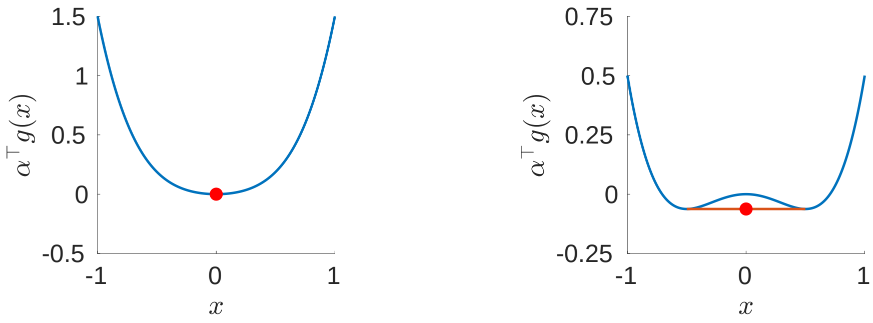
Example 3.21. Let $D := [-1, 1]$, $g_1(x) := x^4$, $g_2(x) := x^4 - x^2$, $\bar{x} = 0$ and $\bar{\alpha} = (1, 0)$. The linear combination of the constraint functions is given by

$$\alpha^\top g = (\alpha_1 + \alpha_2)x^4 - \alpha_2 x^2.$$

We consider a small neighborhood of $\bar{\alpha}$. For $\alpha_2 \leq 0$, we derive that $\alpha^\top g$ is convex and that $f(\alpha) = 0$ holds (see Figure 3.3(a)). For $\alpha_2 > 0$, the extreme points of the minimizing segment are given by the two minima of $\alpha^\top g$, i.e., $x^{1,2}(\alpha) = \pm\sqrt{\frac{\alpha_2}{2(\alpha_1+\alpha_2)}}$ (see Figure 3.3(b)). The value of the convex envelope in this case is therefore $f(\alpha) = -\frac{\alpha_2^2}{4(\alpha_1+\alpha_2)}$. We derive

$$f(\alpha) = \begin{cases} 0, & \text{if } \alpha_2 \leq 0, \\ -\frac{\alpha_2^2}{4(\alpha_1+\alpha_2)}, & \text{if } \alpha_2 > 0. \end{cases}$$

This function is differentiable at $\bar{\alpha} = (1, 0)$. However, the only continuous choice of $x^1(\alpha), \ldots, x^{n+1}(\alpha)$ at $\bar{\alpha}$ is $x^1(\bar{\alpha}) = x^2(\bar{\alpha}) = 0$. These points are not affinely independent.



(a) Convex envelope of $\alpha^\top g$ at $\bar{x}$ for $\alpha_2 \leq 0$. (b) Convex envelope of $\alpha^\top g$ at $\bar{x}$ for $\alpha_2 > 0$.

Figure 3.3: Visualization of Example 3.21.

In this section, we derived a criterion for showing differentiability of $f(\alpha)$ at specific points. However, the proposed Separation Problem 3.15 is in general not differentiable for all points $\alpha \in B^m$ (see Example 3.17). In order to derive a solution of Problem 3.15, we need suitable solution methods for non-differentiable problems. See for example [Lemaréchal, 1989; Mäkelä, 2002; Shor, 2012] for an introduction on this topic. Simple interior point methods work similar to steepest descent methods for convex differentiable problems. In every iteration, a subgradient is used as a descent direction with a fixed diminishing step size. Bundle methods on the other hand aim to approximate the considered function from below. This approximation is iteratively improved by the approximate solution and the respective subgradient. These strategies require the objective value as well as a subgradient for every iteration point $\alpha \in B^m$. Based on Corollary 3.19, both are given directly by any solution of (3.4). This allows us to solve Problem 3.15 if a minimizing simplex is available for all linear combinations of the constraint functions.

3.2.4 Deriving Linear Inequalities

In Section 3.2.3, we discussed the structure of the objective function and possible solution strategies for Separation Problem 3.15. In the following, we assume that a solution to Separation Problem 3.15 can be computed. In order to algorithmically utilize this solution in a cutting plane approach, we need to find a linear inequality that separates the point $(\bar{x}, \bar{z})$ from

$$Y = \operatorname{conv}(X) = \operatorname{conv}\big(\{(x, g(x)) \mid x \in D\}\big).$$

We establish the following notation for our analysis.

Definition 3.22. Let $Z \subseteq \mathbb{R}^n$ be a closed convex set. Furthermore, for $\beta \in \mathbb{R}^n$ and $\beta_0 \in \mathbb{R}$, we consider the linear inequality $\beta^\top z \leq \beta_0$ and the corresponding hyperplane $\mathcal{H}(\beta, \beta_0) := \{z \in \mathbb{R}^n \mid \beta^\top z = \beta_0\}$.

- The inequality $\beta^\top z \leq \beta_0$ is called a valid inequality for Z if $\beta^\top z \leq \beta_0$ holds for all $z \in Z$.
- Let $\bar{z} \in Z$. We call $\mathcal{H}(\beta, \beta_0)$ a supporting hyperplane of Z at $\bar{z}$ if $\beta^\top z \leq \beta_0$ is valid for Z and $\bar{z} \in \mathcal{H}(\beta, \beta_0)$.
- Let $\bar{z} \notin Z$. We call $\mathcal{H}(\beta, \beta_0)$ a cutting plane of Z for $\bar{z}$ if $\beta^\top z \leq \beta_0$ is valid for Z but not valid for $\{\bar{z}\}$.

The respective separation task using cutting planes is defined as

Separation Task 3.23.

`Input`: A compact and convex set $D \subseteq \mathbb{R}^n$, a continuous function $g : D \to \mathbb{R}^m$ and a point $(\bar{x}, \bar{z}) \in D \times \mathbb{R}^m$.

`Task`: Decide whether $(\bar{x}, \bar{z}) \in Y$, and, if not, return a vector $(b, a, b_0) \in \mathbb{R}^{n+m+1}$, such that

$$\mathcal{H}(b, a, b_0) = \big\{(x, z) \in \mathbb{R}^{n+m} \mid b^\top x + a^\top z = b_0\big\}$$

is a cutting plane of Y for $(\bar{x}, \bar{z})$.

Our analysis and algorithmic results are summarized in the following corollary.

Corollary 3.24. *Let $D \subseteq \mathbb{R}^n$ be compact and convex and $g : D \to \mathbb{R}^m$ continuous.*

1. *Let $\alpha \in \mathbb{R}^m$ and $\bar{x} \in D$ be fixed. Let $(\beta, \beta_0) \in \mathbb{R}^{(n+1)+1}$ with $\beta_{n+1} = -1$ define a supporting hyperplane $\mathcal{H}(\beta, \beta_0)$ of $\operatorname{epi}\big(\operatorname{vex}_D[\alpha^\top g], D\big)$ at the point $\big(\bar{x}, \operatorname{vex}_D[\alpha^\top g](\bar{x})\big)$. Then, a valid linear inequality for Y is given by*

$$(\beta_1, \dots, \beta_n, -\alpha)^\top (x, z) \leq \beta_0 \tag{3.6}$$

   We denote $\mathcal{V}_{\alpha,\bar{x}}[\beta, \beta_0] := \mathcal{H}(\beta_1, \dots, \beta_n, -\alpha, \beta_0)$.

2. *Let $(\bar{x}, \bar{z}) \notin Y$ and let $\alpha^\star$ be an optimal solution to Problem 3.15. Let $(\beta, \beta_0) \in \mathbb{R}^{(n+1)+1}$ with $\beta_{n+1} = -1$ define a supporting hyperplane $\mathcal{H}(\beta, \beta_0)$ of $\operatorname{epi}\big(\operatorname{vex}_D[\alpha^{\star\top} g], D\big)$ at the point $\big(\bar{x}, \operatorname{vex}_D[\alpha^{\star\top} g](\bar{x})\big)$. Then $\mathcal{V}_{\alpha^\star,\bar{x}}[\beta, \beta_0]$ is a cutting plane of Y for $(\bar{x}, \bar{z})$.*

Proof.

1. First, we define the half-space given by (3.6) intersected with box D as

$$N_{g,\bar{x}}(\alpha, \beta, \beta_0) := \big\{(x, z) \in \mathbb{R}^{n+m} \mid (\beta_1, \dots, \beta_n, -\alpha)^\top (x, z) \leq \beta_0,\ x \in D\big\}.$$

   $\mathcal{H}(\beta, \beta_0)$ is a supporting hyperplane of $\operatorname{epi}\big(\operatorname{vex}_D[\alpha^\top g]\big)$ at the point $\big(\bar{x}, \operatorname{vex}_D[\alpha^\top g](\bar{x})\big)$. For every $(x, \hat{z}) \in \mathcal{H}(\beta, \beta_0)$ it follows

$$\hat{z} \leq \operatorname{vex}_D[\alpha^\top g](x)$$

   and

$$\hat{z} = \frac{\beta_0 - \sum_{i=1}^n \beta_i x_i}{\beta_{n+1}} = \sum_{i=1}^n \beta_i x_i - \beta_0.$$

   For any point $(x, z) \in M_g(\alpha)$ we have

$$\begin{aligned} & -\alpha^\top z + \operatorname{vex}_D[\alpha^\top g](x) \leq 0 \\ \Rightarrow \quad & -\alpha^\top z + \sum_{i=1}^n \beta_i x_i - \beta_0 \leq 0 \end{aligned}$$

and therefore $(x,z) \in N_{g,\bar{x}}(\alpha,\beta,\beta_0)$. This implies $M_g(\alpha) \subseteq N_{g,\bar{x}}(\alpha,\beta,\beta_0)$, and that the linear inequality

$$(\beta_1,\dots,\beta_n,-\alpha)^\top(x,z) \le \beta_0$$

is valid for Y.

2. Let $(\bar{x},\bar{z}) \notin Y$. Using Observation 3.16 we obtain

$$\alpha^{\star\top}\bar{z} < \operatorname{vex}_D[\alpha^{\star\top} g](\bar{x}).$$

As $\mathcal{H}(\beta,\beta_0)$ is a supporting hyperplane at $\big(\bar{x}, \operatorname{vex}_D[\alpha^{\star\top} g](\bar{x})\big)$, we have

$$\big(\bar{x}, \operatorname{vex}_D[\alpha^{\star\top} g](\bar{x})\big) \in \mathcal{H}(\beta,\beta_0).$$

We follow the proof of 1. and derive

$$\sum_{i=1}^{n} \beta_i \bar{x}_i - \beta_0 = \operatorname{vex}_D[\alpha^{\star\top} g](\bar{x}).$$

Combining these equations, we have

$$\alpha^{\star\top}\bar{z} < \sum_{i=1}^{n} \beta_i \bar{x}_i - \beta_0.$$

This leads to $(\bar{x},\bar{z}) \notin N_{g,\bar{x}}(\alpha^\star,\beta,\beta_0)$ and to our statement.

□

Remark 3.25. Note that the restriction $\beta_{n+1} = -1$ on the supporting hyperplane of epi $\big(\operatorname{vex}_D[\alpha^\top g], D\big)$ is not a strong restriction. In fact, there is no supporting hyperplane $\mathcal{H}(\beta,\beta_0)$ with $\beta_{n+1} > 0$ of $\operatorname{epi}(f,D)$ for any continuous $f : D \to \mathbb{R}$. This holds as $(x,z) \in \operatorname{epi}(f,D)$ leads to $(x, z+\varepsilon) \in \operatorname{epi}(f,D)$ for all $\varepsilon \ge 0$. Furthermore, the case $\beta_{n+1} = 0$ only occurs for $x \in \operatorname{bd}(D)$. For $\beta_{n+1} < 0$ finally, we may assume $\beta_{n+1} = -1$ without loss of generality.

A direct consequence of Corollary 3.24 is that we can find an exact representation of Y based on supporting hyperplanes of the epigraph of the convex envelopes for different α (see also [Ballerstein, 2013]). For this, let $\big(\beta(x,\alpha), \beta_0(x,\alpha)\big) \in \mathbb{R}^{(n+1)+1}$ with $\beta_{n+1}(x,\alpha) = -1$ define an arbitrary supporting hyperplane $\mathcal{H}\big(\beta(x,\alpha),\beta_0(x,\alpha)\big)$ of epi $\big(\operatorname{vex}_D[\alpha^\top g]\big)$ at the point $\big(x, \operatorname{vex}_D[\alpha^\top g](x)\big)$. Then we have

$$Y = \bigcap_{\alpha \in B^m} \Big(\bigcap_{x \in D} N_g\big(\alpha, \beta(x,\alpha), \beta_0(x,\alpha)\big) \Big).$$

Concluding these results, we are able to design cutting planes of the set Y by solving a convex, non-differentiable optimization problem. For this, we need an algorithmically utilizable representation of $\mathrm{vex}_D[\alpha^\top g](\bar{x})$ as well as a supporting hyperplane at a given point $\bar{x}$ for arbitrary $\alpha \in B^m$.

These requirements are quite high and usually not satisfiable for general constraint sets. In Chapter 4, we focus on a special constraint consisting of bivariate quadratic absolute value functions. For this case, we derive the convex envelope and make use of the proposed cutting planes in order to show improvements of the quality of the relaxed feasible set.

The remainder of this chapter is used to generalize the prior analysis. We derive a weaker separation result that can be applied with only an estimation of the convex envelope.

3.3 Estimating the Convex Envelope

Recall that the proposed separation strategy in Section 3.2 is based on Separation Problem 3.15. The main drawback of this problem is its dependency on the convex envelope, that is required for all linear combinations of the constraint functions. As the convex envelope is in general not available for arbitrary functions, we relax this condition in the following. To be more precise, we substitute the convex envelope in the proposed Separation Problem 3.15 by a convex underestimator. We discuss the required properties of such an underestimator and present the respective results for the separation strategy. Furthermore, we derive ways of constructing an underestimator that meets the requirements.

3.3.1 Sufficient Criteria for Separation

We consider the setting given in Section 3.2 with a continuous vector-valued constraint function $g : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$, a feasible set

$$X = \big\{(x, z) \mid z = g(x), x \in D\big\}$$

and its convex hull $Y = \mathrm{conv}(X)$.

Recall that Separation Problem 3.15 is given as

$$\min_{\alpha \in B^m} h(\alpha) := \alpha^\top \bar{z} - \mathrm{vex}_D[\alpha^\top g](\bar{x}).$$

In this section, we aim to replace the convex envelope by a convex underestimator. On the one hand, it is usually a lot easier to construct a convex underestimator instead of the convex envelope. On the other hand, as the convex envelope is just a special underestimator, we still expect similar results considering the proposed separation strategy.

However, as the convex underestimator for a given function is not uniquely defined, we first introduce a suitable selection of estimators. For this, consider the set of relevant functions (see Problem 3.15) given as

$$F := \{\alpha^\top g \mid \alpha \in B^m\}.$$

The selection of estimators can now be interpreted as a function itself, that simply maps every $f \in F$ onto an estimator of f. We formalize this concept by the following definition.

Definition 3.26. Let $D \subseteq \mathbb{R}^n$ be compact and convex. Let $C^D(\mathbb{R})$ be the space of continuous functions mapping from D to $\mathbb{R}$. Let $E \subseteq C^D(\mathbb{R})$ be convex. We call a function $\sigma : E \to C^D(\mathbb{R})$, $f \mapsto \sigma[f]$ an estimator selection of E on D.

We further introduce two suitable properties of estimator selections. The first one ensures that the estimator selection always maps onto a convex underestimator of the considered function. The second one is motivated by the property of the convex envelope presented in Proposition 3.16. It is designed to ensure the convexity of the resulting separation problem.

Definition 3.27. Let $D \subseteq \mathbb{R}^n$ be compact and convex, and let $E \subseteq C^D(\mathbb{R})$ convex.

- We call an estimator selection of E on D a convex underestimator selection of E on D if $\sigma[f]$ is a convex underestimator of f on D for every $f \in E$.
- We call a (convex under-) estimator selection σ of E on D consistent if

  $$\lambda\sigma[f_1](x) + (1-\lambda)\sigma[f_2](x) \leq \sigma\big[\lambda f_1 + (1-\lambda) f_2\big](x)$$

  holds for all $f_1, f_2 \in E$, every $\lambda \in [0, 1]$ and all $x \in D$.

Remark 3.28. The last property of σ being consistent is equivalent to σ being concave on E. However, we avoid this expression in order to prevent confusion. For a consistent convex underestimator selection σ of F on D, it follows that $\sigma : E \to C^D(\mathbb{R})$ is concave but that $\sigma[f] : D \to \mathbb{R}$ is convex for all $f \in E$.

Example 3.29. Consider $\sigma^\star : C^D(\mathbb{R}) \to C^D(\mathbb{R})$ with $\sigma^\star[f] := \operatorname{vex}_D[f]$ that maps every function onto its convex envelope. By definition and Proposition 3.16, $\sigma^\star$ is a consistent convex underestimator selection of $C^D(\mathbb{R})$ on D. Furthermore,

$$\bar{\sigma}[f](x) \leq \sigma^\star[f](x)$$

holds for all convex underestimator selections $\bar{\sigma}$ of $E \subseteq C^D(\mathbb{R})$ on D, $f \in E$ and $x \in D$.

Now, let σ be an estimator selection of $F = \left\{\alpha^\top g \mid \alpha \in B^m\right\}$ on D. We consider a given point $(\bar{x}, \bar{z})$ and adapt Separation Problem 3.15 to derive the Approximate Problem given as

Problem 3.30.

$$\min_{\alpha \in B^m} h(\alpha) := \alpha^\top \bar{z} - \sigma[\alpha^\top g](\bar{x}).$$

Using Definition 3.26 and 3.27, we directly generalize the prior results.

Proposition 3.31.

1. *The objective function $h : \mathbb{R}^m \to \mathbb{R}$ is continuous.*

2. *Let $\alpha \in \mathbb{R}^m$ with $h(\alpha) < 0$. If σ is a convex underestimator selection, then $(\bar{x}, \bar{z}) \notin M_g(\alpha)$ and therefore $(\bar{x}, \bar{z}) \notin Y$.*

3. *If σ is consistent, then the Approximate Problem 3.30 is convex.*

Proof.

1/2. By construction.

3. See proof of Proposition 3.16

□

This means that choosing a consistent convex underestimator selection σ of F on D in Problem 3.30 provides two important benefits. First of all, Problem 3.30 is convex and can therefore be solved to global optimality. This observation is based on σ being consistent. Second, the optimal value provides a sufficient condition that the optimal solution is suitable for separating a given point $(\bar{x}, \bar{z})$ from Y. This condition is based on σ being a convex underestimator selection. However, as $\sigma[f](\bar{x}) < \text{vex}_D[f](\bar{x})$ holds in general for $f \in F$, we do not derive a necessary condition anymore.

In a next step, we present ways of designing consistent convex underestimator selections. This topic is worth to discuss, as several standard options for convex underestimators do not lead to suitable estimator selections.

3.3.2 Design of Estimator Selections

We aim to derive suitable estimator selections to apply Proposition 3.31. As a motivation, we use the following example to show that the property of being consistent is not given automatically by convex underestimator selections.

Example 3.32. It is common to construct algorithmically utilizable convex underestimators for convex functions as the maximum of tangents. We consider two convex functions $g_1(x) = x^2$ and $g_2(x) = x^4$ on $[0, 1]$. Let the (convex under-) estimator selection σ be defined as follows. For every function f, $\sigma[f]$ is given as

$$\sigma[f](x) = \max\big(f(y) + \nabla f(y)^\top (x - y), f(z) + \nabla f(z)^\top (x - z)\big)$$

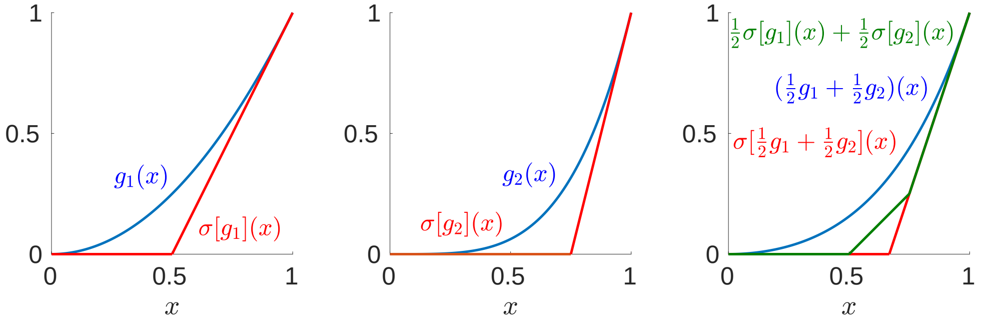
with $y = 0$ and $z = 1$.

Below, we display g_1 and $\sigma[g_1]$ in Figure 3.4(a), and g_2 and $\sigma(g_2)$ in Figure 3.4(b). Figure 3.4(c) shows the convex combinations $\frac{1}{2}(g_1+g_2), \sigma\left[\frac{1}{2}(g_1+g_2)\right]$ and $\frac{1}{2}\sigma[g_1](x) + \frac{1}{2}\sigma[g_2]$. Obviously,

$$\frac{1}{2}\sigma[g_1](x) + \frac{1}{2}\sigma[g_2](x) > \sigma\Big[\frac{1}{2}g_1 + \frac{1}{2}g_2\Big](x)$$

holds for $x \in (\frac{1}{2}, \frac{3}{4})$. Therefore, σ is not a consistent estimator selection.

Note that the presented interaction is implicitly already studied in [Tawarmalani and Sahinidis, 2005]. The authors show that a tighter convex underestimator of a composition of convex functions can be designed by considering the convex underestimators of the single functions.



(a) Graph of g_1 and $\sigma[g_1]$. (b) Graph of g_2 and $\sigma[g_2]$. (c) Convex combinations.

Figure 3.4: Visualization of Example 3.32.

The motivation, why the kind of estimator selection used in Example 3.32 is not consistent, lies in the fact that the resulting function is given as a maximum of certain values. It is well known that

$$\max(a_1 + b_1, a_2 + b_2) \leq \max(a_1, a_2) + \max(b_1, b_2)$$

holds. In the case of Example 3.32, a_1, b_1 represent the tangents at y and a_2, b_2 the ones at z. The convex envelope on the other hand is given as the minimum of certain values. To be explicit, it is given as the minimum of all suitable convex combinations of points in the graph. For the minimum operator we have

$$\min(a_1 + b_1, a_2 + b_2) \geq \min(a_1, a_2) + \min(b_1, b_2),$$

which results in the desired property of $\sigma^\star$ being consistent.

This observation gives a good intuition that consistent estimator selections should be designed in a similar way as the convex envelope. In the following, we present an estimator selection that is based on computing the convex envelope only w.r.t. to a discretized set of points.

Adjusted Discretization

Let D be a polytope and let $G \subseteq D$ be a discretization of D, i.e., a finite set with $D = \operatorname{conv}(G)$. For any $g : G \to \mathbb{R}$, we define $\sigma_G[g] : D \to \mathbb{R}$ point-wise by a simplified version of the problem used to derive the convex envelope.

$$\begin{aligned} \sigma_G[g](\bar{x}) := \min & \sum_{i=1}^{n+1} \lambda_i \cdot g(x^i) \\ \text{s.t.} & \sum_{i=1}^{n+1} \lambda_i \, x^i = \bar{x} \\ & \sum_{i=1}^{n+1} \lambda_i = 1 \\ & \lambda_i \geq 0, \quad x^i \in G, \quad i = 1, \dots, n+1. \end{aligned} \tag{3.7}$$

This problem is feasible because of $D = \operatorname{conv}(G)$. The image $\sigma_G[g]$ may also be interpreted as a convex extension of g (see [Tawarmalani and Sahinidis, 2001]) or as a "konvexe Unterfunktion" of g (see [Kleibohm, 1967]) on a subset $G \subseteq D$.

In this context however, we use function σ_G as an estimator selection of $C^D(\mathbb{R})$ on D and derive the following desirable properties.

Lemma 3.33.

1. *The problem in* (3.7) *reduces to a linear one.*

2. *The function $\sigma_G[g]$ is convex for every $g : G \to \mathbb{R}$.*

3. *The estimator selection σ_G is consistent.*

Proof.

1. With a finite set G, the points $x^i \in G$ may be interpreted as parameters and the only variables are λ_i for $i = 1, \dots, |G|$. By Caratheodory's Theorem, there always exists an optimal solution $\lambda^\star \in \mathbb{R}^{|G|}$ with only $n+1$ non-zero components. These non-zero components, together with their respective points x^i, give a solution to (3.7).

2. Consider two points $\bar{x}, \bar{y} \in D$ with their respective optimal solutions of (3.7) for $\sigma_G[g](\bar{x})$ and $\sigma_G[g](\bar{y})$ given by $(\lambda; x^1, \dots, x^{n+1})$ and $(\mu; y^1, \dots, y^{n+1})$. The convex combination of $\bar{x}$ and $\bar{y}$ is denoted by $\bar{z} := \nu\bar{x} + (1-\nu)\bar{y}$ with $\nu \in [0,1]$. We reformulate it into

$$\bar{z} = \nu\bar{x} + (1-\nu)\bar{y} = \sum_{i=1}^{n+1} \nu\lambda_i x^i + \sum_{i=1}^{n+1} (1-\nu)\mu_i y^i,$$

with $x^1, \dots, x^{n+1}, y^1, \dots, y^{n+1} \in G$ and $\sum_{i=1}^{n+1} \nu\lambda_i + \sum_{i=1}^{n+1}(1-\nu)\mu_i = 1$. We denote the convex combination of the respective objective values by

$$\bar{g} := \sum_{i=1}^{n+1} \nu\lambda_i \cdot g(x^i) + \sum_{i=1}^{n+1}(1-\nu)\mu_i \cdot g(y^i).$$

Now we interpret $(\bar{z}, \bar{g})$ as a convex combination of points

$$\big(x^i, g(x^i)\big) \in \operatorname{epi}(g, G) \quad \text{and} \quad \big(y^i, g(y^i)\big) \in \operatorname{epi}(g, G),$$

and derive $(\bar{z}, \bar{g}) \in \operatorname{conv}\big(\operatorname{epi}(g, G)\big)$. It is either $(\bar{z}, \bar{g})$ on the boundary of $\operatorname{conv}\big(\operatorname{epi}(g, G)\big)$, or there exist some $(\bar{z}, g^\star) \in \operatorname{conv}\big(\operatorname{epi}(g, G)\big)$ with $g^\star \leq \bar{g}$. By Caratheodory's Theorem, this means that there exist points $z^1, \dots, z^{n+1} \in D$ and coefficients $\gamma \in [0,1]^{n+1}$ with

$$\sum_{i=1}^{n+1} \gamma_i \cdot g(z^i) \leq \bar{g},$$
$$\sum_{i=1}^{n+1} \gamma_i \, z^i = \bar{z},$$
$$\sum_{i=1}^{n+1} \gamma_i = 1.$$

Hence, $(\gamma; z^1, \dots, z^{n+1})$ is a feasible solution of (3.7) for $\sigma_G[g](\bar{z})$. The associated objective value is given by $\sum_{i=1}^{n+1} \gamma_i g(z^i)$ while the optimal value is $\sigma_G[g]\big(\nu\bar{x} + (1-\nu)\bar{y}\big)$. We conclude

$$\begin{aligned}
\sigma_G[g]\big(\nu\bar{x} + (1-\nu)\bar{y}\big) &\leq \sum_{i=1}^{n+1} \gamma_i g(z^i) \\
&\leq \bar{g} \\
&= \sum_{i=1}^{n+1} \nu\lambda_i \cdot g(x^i) + \sum_{i=1}^{n+1}(1-\nu)\mu_i \cdot g(y^i) \\
&= \nu\sigma_G[g](\bar{x}) + (1-\nu)\sigma_G[g](\bar{y}).
\end{aligned}$$

The statement follows.

3. Similar to the convex envelope, the function $\sigma_G[g]$ can be interpreted as the point-wise supremum over all convex functions that have a lower value at the points of the discretization, i.e.,

$$\sigma_G[g](\bar{x}) = \sup\big\{h(\bar{x}) \mid h(x) \leq g(x) \;\forall\; x \in G,\; h \text{ convex}\big\}.$$

For the property of being consistent, let $g_1, g_2 \in C^D(\mathbb{R})$. Obviously, $\lambda\sigma_G[g_1] + (1-\lambda)\sigma_G[g_2]$ is convex and

$$\begin{aligned}\lambda\sigma_G[g_1](x) + (1-\lambda)\sigma_G[g_2](x) &\leq \lambda g_1(x) + (1-\lambda)g_2(x)\\ &= \big(\lambda g_1 + (1-\lambda)g_2\big)(x)\end{aligned}$$

holds for all $x \in G$. This means that $\sigma_G\big[\lambda g_1 + (1-\lambda)g_2\big]$ is the supremum of a set that includes $\lambda\sigma_G[g_1] + (1-\lambda)\sigma_G[g_2]$. Hence, we conclude

$$\lambda\sigma_G[g_1] + (1-\lambda)\sigma_G[g_2] \leq \sigma_G\big[\lambda g_1 + (1-\lambda)g_2\big]$$

and σ is consistent.

□

However, the behavior of the input function between the discretized points is not considered in the approach above. Therefore, $\sigma_G[g] \leq g$ does not hold in general and σ_G is not a convex underestimator selection. In order to account for this behavior, we adjust every function value at the discretized points based on the first derivative and the size of the set D. As a first step, the following theorem will allow us to estimate the difference between the values of g and $\sigma_G[g]$.

Theorem 3.34. *Let $D \subseteq \mathbb{R}^n$ and $G := \{x^1, \dots, x^m\} \subseteq \mathbb{R}^n$ with $m = n+1$ and $\operatorname{conv}(G) = D$. Let $g : D \to \mathbb{R}$ be differentiable with $||\nabla g(x)||_2 \leq R$ for all $x \in D$. Then*

$$g(x) \geq \sigma_G[g](x) - R\frac{n}{n+1}d^\star$$

holds for all $x \in D$, where

$$d^\star := \operatorname{diam}(G) := \max_{x,y\in G} ||x-y||$$

is the diameter of G.

Proof. Consider an arbitrary point $x \in D$. We denote the distance of x to the points in G by

$$d_i := ||x - x^i||_2$$

for all $i = 1, \dots, m$. Using the mean value theorem, there exist some $y^i \in D$ with

$$\begin{aligned} g(x) &= g(x^i) + \nabla g(y^i) \cdot (x - x^i) \\ &\geq g(x^i) - ||\nabla g(y^i)||_2 \cdot ||(x - x^i)||_2 \\ &\geq g(x^i) - R \cdot d_i \end{aligned}$$

for $i = 1, \dots, m$. Let λ_i be given by any optimal solution of (3.7) for $\sigma_G[g](x)$, i.e., $\sum_{i=1}^m \lambda_i \, x^i = x$, $\sum_{i=1}^m \lambda_i = 1$ and $\lambda_i \geq 0$ for all $i = 1, \dots, m$ with $m = n+1$. Using the non-negativity of λ, we derive

$$\begin{aligned} & \sum_{i=1}^m \lambda_i g(x) \geq \sum_{i=1}^m \lambda_i g(x^i) - R \sum_{i=1}^m \lambda_i d_i \\ \Leftrightarrow \qquad & g(x) \geq \sigma_G[g](x) - R \sum_{i=1}^m \lambda_i d_i . \end{aligned}$$

Next, we estimate the term $\lambda_j d_j$ for every fixed $j \in 1, \dots, m$. We assume $\lambda_j \neq 0$ and $\lambda_j \neq 1$, as the term is equal to zero otherwise. This follows for $\lambda_j = 0$ trivially and for $\lambda_j = 1$ because of $x = x^j$ and therefore $d_j = 0$. We set

$$\begin{aligned} \bar{\lambda}_j &:= \sum_{i=1,\, i \neq j}^m \lambda_i = 1 - \lambda_j , \\ \bar{x}^j &:= \sum_{i=1,\, i \neq j}^m \frac{\lambda_i}{\bar{\lambda}_j} x^i , \\ \bar{d}_j &:= ||x - \bar{x}^j||_2 \end{aligned}$$

with $\bar{x}^j$ being a convex combination of points in G, and therefore with $\bar{x}^j \in D$. Furthermore, we interpret x as a convex combination of the two points x^j and $\bar{x}^j$ by

$$x = \sum_{i=1}^m \lambda_i x^i = \lambda_j x^j + \sum_{i=1, i \neq j}^m \lambda_i x^i = \lambda_j x^j + \bar{\lambda}_j \sum_{i=1, i \neq j}^m \frac{\lambda_i}{\bar{\lambda}_j} x^i = \lambda_j x^j + \bar{\lambda}_j \bar{x}^j .$$

This implies the following:

1. The point x lies on the line connecting x^j and $\bar{x}^j$. Hence, we have

$$d_j + \bar{d}_j = ||x^j - x||_2 + ||\bar{x}^j - x||_2 = ||x^j - \bar{x}^j||_2 \leq d^\star .$$

The inequality holds, as the diameter of D is the same as the diameter of G.

2. In the two dimensional case, the coefficient λ_j behaves inversely proportional to d_j. It is

$$\lambda_j = \frac{\bar{d}_j}{d_j + \bar{d}_j} \quad \text{and} \quad \bar{\lambda}_j = \frac{d_j}{d_j + \bar{d}_j}.$$

3. Combining all of the above, we derive

$$\lambda_j d_j = \lambda_j \bar{\lambda}_j (d_j + \bar{d}_j) \leq \lambda_j (1 - \lambda_j) d^\star.$$

Hence, the difference between function g and $\sigma_G[g]$ can be estimated by

$$g(x) \geq \sigma_G[g](x) - Rd^\star \sum_{i=1}^{m} \lambda_i (1 - \lambda_i). \tag{3.8}$$

We further simplify this expression by deriving the maximum of the sum. We consider

$$\begin{aligned} \max\ f(\lambda) &:= \sum_{i=1}^{m} \lambda_i (1 - \lambda_i) \\ \text{s.t.}\quad h(\lambda) &:= \sum_{i=1}^{m} \lambda_i - 1 = 0. \end{aligned}$$

The problem is convex and differentiable, so we apply KKT conditions. For $\lambda^\star = \frac{1}{m} \sum_{i=1}^{m-1} e_i$ and $\mu = -\frac{2-m}{m}$, we obtain

$$\nabla f(\lambda^\star) + \mu \nabla h(\lambda^\star) = 0.$$

This leads to the optimal point $\lambda^\star$ and the optimal value

$$f(\lambda^\star) = \sum_{i=1}^{m} \frac{1}{m} \left(1 - \tfrac{1}{m}\right) = 1 - \frac{1}{m} = \frac{m-1}{m}.$$

Using (3.8) and $m = n + 1$, we derive the statement

$$g(x) \geq \sigma_G(g)(x) - R \frac{n}{n+1} d^\star.$$

□

Remark 3.35. The statement in Theorem 3.34 can be generalized for an arbitrary polytope D and $m \geq n + 1$. For any $x \in D$, let $G' := (x^1, \ldots, x^k)$ be the set of the corresponding solution of (3.7) and let $D' := \text{conv}(G')$. Then we have $\sigma_{G'}[g](x) = \sigma_G[g](x)$ and the bound $d^\star$ on the diameter of G, as well as the bound R on the gradient in D, also hold for G' and D'. The statement follows for $m \geq n + 1$.

For a large set D, the error parameter $d^\star$ is accordingly high. This problem can be handled by generating subdivisions of D and by applying the result on every one of them.

Corollary 3.36. *Consider a polytope $D \subseteq \mathbb{R}^n$ and a discretization $G = \{x^1, \dots, x^m\} \subseteq D$ with $m \geq n+1$. Furthermore, let $G_1, \dots, G_l \subseteq G$ with $D_j := \operatorname{conv}(G_j)$ for $j = 1, \dots, l$ and $D \subseteq \bigcup_{j=1}^{l} D_j$. Let $||\nabla g(x)||_2 \leq R_j$ for all $x \in D_j$ and let $d_j^\star = \operatorname{diam}(D_j)$ for all $j = 1, \dots, l$. Then*

$$g(x) \geq \sigma_G[g](x) - R_j d_j^\star \frac{n}{n+1}$$

holds for every $j = 1, \dots, l$ and $x \in D_j$.

Proof. Let $x \in D_j$. Using Theorem 3.34, we obtain

$$g(x) \geq \sigma_{G_j}[g](x) - R_j d_j^\star \frac{n}{n+1}.$$

Furthermore, we have $G_j \subseteq G$ and $x \in D_j \subseteq D$. Hence, the optimal solution of (3.7) for $\sigma_{G_j}[g](x)$ is a feasible solution of (3.7) for $\sigma_G[g](x)$. Therefore, we have $\sigma_G[g](x) \leq \sigma_{G_j}[g](x)$ and our statement

$$g(x) \geq \sigma_G[g](x) - R_j d_j^\star \frac{n}{n+1}.$$

□

The error estimation in Corollary 3.36 is better than the one given in Theorem 3.34. In fact, $d_j^\star \leq d^\star$ and $R_j \leq R$ hold for all $j = 1, \dots, l$, and in general even $d_j^\star < d^\star$ and $R_j < R$ for most $j = 1, \dots, l$.

Now, we are able to design a consistent convex underestimator selection of any $F \subseteq C^D(\mathbb{R})$ on $D \in \mathbb{R}^n$. It relies on a discretization of D and an estimation of the gradient. We denote it by *adjusted discretization* and present two different approaches in the following. The first one is based on a constant bound of the gradient for the entire sets F and D. The second one considers the gradient and the size of the subdivision in a more flexible way.

Corollary 3.37. *Consider a polytope $D \subseteq \mathbb{R}^n$ and a discretization $G = \{x^1, \dots, x^m\} \subseteq D$ with $m \geq n+1$. Furthermore, let $G_1, \dots, G_l \subseteq G$ with $D_j = \operatorname{conv}(G_j)$ for $j = 1, \dots, l$ and $D \subseteq \bigcup_{j=1}^{l} D_j$. Let $d^\star \geq \operatorname{diam}(D_j)$ for all $j = 1, \dots, l$. Let $F \subseteq C^D(\mathbb{R})$ with $||\nabla g(x)||_2 \leq R$ for all $g \in F$ and $x \in D$.*

We set $\bar{g} : G \to \mathbb{R}$ by

$$\bar{g}(x) := g(x) - Rd^\star \frac{n}{n+1}.$$

Then, $\rho : F \to C^D(\mathbb{R})$ with $\rho_G[g] := \sigma_G[\bar{g}]$ is a consistent convex underestimator selection of F on D.

Proof. The term $Rd^\star \frac{n}{n+1}$ does not depend on x and g. With respect to (3.7), any constant term in g may be considered separately because of $\sum_{i=1}^{k} \lambda_i = 1$. Therefore, we have

$$\rho_G[g](x) = \sigma_G[g](x) - Rd^\star \frac{n}{n+1}.$$

Following Lemma 3.33, ρ_G is consistent and $\rho_G[g]$ is convex for all $g \in F$. By Corollary 3.37 and the definition of R and $d^\star$, we also derive

$$\rho_G[g](x) \leq g_G(x).$$

The statement follows. □

The bound R on $||\nabla g(x)||_2$ may be unreasonable high for most subsets D_j of D. Furthermore, the general bound $d^\star$ on all diameters does not allow for a dynamic discretization. The second presented estimator selection takes both problems into account.

Theorem 3.38. *Consider a polytope $D \subseteq \mathbb{R}^n$ and a discretization $G = \{x^1, \ldots, x^m\} \subseteq D$ with $m \geq n+1$. Furthermore, let $G_1, \ldots, G_l \subseteq G$ with $D_j = \operatorname{conv}(G_j)$ for $j = 1, \ldots, l$ and $D \subseteq \bigcup_{j=1}^{l} D_j$. Let $F \subseteq C^D(\mathbb{R})$ and $R_j(g) := \max_{x \in D} ||\nabla g(x)||_2$ for all $x \in D_j$, $g \in F$ and $d_j^\star = \operatorname{diam}(D_j)$ for all $j = 1, \ldots, l$. We set $\bar{g} : G \to \mathbb{R}$ by*

$$\bar{g}(x^i) = g(x^i) - \varepsilon_i(g) \frac{n}{n+1}$$

with

$$\begin{aligned} \varepsilon_i(g) := \max\ & R_j(g) d_j^\star \\ \text{s.t.}\ \ & x^i \in G_j \\ & j \in \{1, \ldots, l\}. \end{aligned}$$

Then, a consistent convex underestimator selection of F on D is given by $\tau_G : F \to C^D(\mathbb{R})$ with $\tau_G[g] := \sigma_G[\bar{g}]$.

Proof. Using Lemma 3.33 and Corollary 3.37, $\tau_G[g]$ is convex and $\tau_G[g](x) \leq g(x)$ holds for all $g \in F$ and all $x \in D$. In order to show that τ is consistent, we interpret $\tau_G[g]$ as the supremum over all convex functions with

$$\tau_G[g](x^i) \leq g(x^i) - \varepsilon_i(g)\frac{n}{n+1}$$

for every $i = 1, \dots, m$ (similar to Lemma 3.33). Then, with $\lambda \in [0,1]$ and $g_1, g_2 \in F$, we have

$$\begin{aligned}
&\lambda\tau_G[g_1](x^i) + (1-\lambda)\tau_G[g_2](x^i) \\
\leq\ & \lambda\left(g_1(x^i) - \varepsilon_i(g_1)\frac{n}{n+1}\right) + (1-\lambda)\left(g_2(x^i) - \varepsilon_i(g_2)\frac{n}{n+1}\right) \\
=\ & \big[\lambda g_1 + (1-\lambda)g_2\big](x^i) - \big(\lambda\varepsilon_i(g_1) + (1-\lambda)\varepsilon_i(g_2)\big)\frac{n}{n+1}
\end{aligned}$$

for all $i = 1, \dots, m$. The term $\varepsilon_i(g)$ can be interpreted as the maximal absolute gradient of

$$h(x) := g(x) \cdot d^\star(x) \qquad \text{with} \qquad d^\star(x) := \max_{j=1,\dots,l,\ x \in D_i} d_j^\star.$$

The gradient of h behaves linear in g and the maximal gradient of the sum of two functions is smaller than the sum of the maximal gradients of two functions. In other words, $\varepsilon_i(g)$ is concave. Therefore,

$$\begin{aligned}
&\lambda\tau_G[g_1](x^i) + (1-\lambda)\tau_G[g_2](x^i) \\
\leq\ & \big[\lambda g_1 + (1-\lambda)g_2\big](x^i) - \big(\lambda\varepsilon_i(g_1) + (1-\lambda)\varepsilon_i(g_2)\big)\frac{n}{n+1} \\
\leq\ & \big[\lambda g_1 + (1-\lambda)g_2\big](x^i) - \varepsilon_i\big(\lambda g_1 + (1-\lambda)g_2\big)\frac{n}{n+1}
\end{aligned}$$

holds for all $i = 1, \dots, m$. As $\lambda\tau_G[g_1](x^i) + (1-\lambda)\tau_G[g_2](x^i)$ is convex, and $\tau_G[\lambda g_1 + (1-\lambda)g_2]$ is the supremum over all convex functions with the property shown above, we derive

$$\lambda\tau_G[g_1](x) + (1-\lambda)\tau_G[g_2](x) \leq \tau_G\big[\lambda g_1 + (1-\lambda)g_2\big](x)$$

for all $g_1, g_2 \in F$ and $x \in D$. Hence, τ_G is a consistent convex underestimator selection of F on D. □

Remark 3.39. The statement in Theorem 3.38 also holds for different choices of R_j. The most relevant choice is a constant R as a boundary for the gradient on the whole sets F and D, as done in Corollary 3.37. This allows for a flexible discretization without the need of computing the maximal gradient in every subset D_j. Choosing R_j as the norm of the element-wise maximum of the gradient does also result in a consistent convex underestimator selection. This value can be determined more easily in general.

Summarizing the results, we may substitute the convex envelope in our proposed Separation Problem 3.15 by any consistent convex underestimator selection and still obtain a necessary condition for our separation strategy to work. One possible estimator selection with the desired properties is the adjusted discretization. Given a discretization and bounds on the gradient of the considered functions, we are able to compute the value of the estimator selection by a linear problem. The number of variables in this problem is equal to the number of discretization points. The adjusted discretization has a similar form as the convex envelope, so that the resulting separation problem is again not differentiable in general. However, according to Corollary 3.19 we are at least able to derive a subgradient directly from an optimal solution of the linear problem. In the following subsection, we discuss some relevant function classes for which an estimation of the gradient can be computed efficiently.

Adjusted Discretization for Special Functions

Again, we consider a feasible set given as the graph of a vector-valued function. In this special case, each entry is given by the same function type and varies only in its argument. For instance, let

$$X := \big\{(x, y, z) \in \mathbb{R}^5 \mid z_1 = f(x), z_2 = f(y), z_3 = f(x + y), (x, y) \in D\big\}$$

with $D := [l, u] \subseteq \mathbb{R}^2$ and some $f : \mathbb{R} \to \mathbb{R}$. In particular, we consider monomials as a common type of one dimensional basic functions, i.e., $f(x) := x^p$ with $p \in \mathbb{N}$.

Our aim is to separate from $Y := \operatorname{conv}(X)$. As the convex envelope of

$$f_\alpha(x, y) := \alpha_1 f(x) + \alpha_2 f(y) + \alpha_3 f(x + y)$$

is not known for general α and p, we are not able to apply Separation Problem 3.15. Instead, we make use of the Approximate Problem 3.30. For this, we need a consistent convex underestimator selection of F on D with

$$F := \left\{\alpha_1 f(x) + \alpha_2 f(y) + \alpha_3 f(x+y) \mid \alpha \in B^3\right\}.$$

We choose the adjusted discretization presented above, and briefly discuss the two options based on Corollary 3.37 and Theorem 3.38.

In order to apply Corollary 3.37, we require a bound on the gradient for all $g \in F$ on D. One possible bound is the elementwise supremum of the absolute gradient. As F is given as a linear combination and the gradient behaves linearly, we may consider the three functions $f(x), f(y), f(x+y)$ separately. The respective gradients are

$$\begin{aligned} \nabla f(x) &= \begin{pmatrix} px^{p-1} \\ 0 \end{pmatrix}, \\ \nabla f(y) &= \begin{pmatrix} 0 \\ py^{p-1} \end{pmatrix}, \\ \nabla f(x+y) &= \begin{pmatrix} p(x+y)^{p-1} \\ p(x+y)^{p-1} \end{pmatrix}. \end{aligned}$$

It is easy to see that

$$\begin{aligned} \nabla f(cx) &= c^{p-1} \nabla f(x), \\ \nabla f(cy) &= c^{p-1} \nabla f(y), \\ \nabla f(c(x+y)) &= c^{p-1} \nabla f(x+y) \end{aligned}$$

hold. Therefore, the absolute maximum of each entry of the gradient is attained at the boundary of the box D and can be derived as the root of a polynomial of degree $p-1$ on each of the four facets. This root can either be determined analytically for small p, or computed numerically for larger p. This procedure is not very time consuming either way. The subdivision $G_1, \ldots, G_l$ of D can be done arbitrarily. The bound $d^\star$ on the diameter is given directly by G_j for $j = 1, \ldots, l$. As a result, we are able to apply Corollary 3.37 and to design the consistent convex underestimator selection ρ of F on D.

In order to apply Theorem 3.38 and to construct the respective τ, we first need a subdivision of D. The set D is a box, so we are able to design G and a subdivision of D that also consists of boxes. According to Remark 3.39, we consider the elementwise supremum of the absolute gradient of a given function on every sub-box. For every $g \in F$, it also holds

$$\nabla g(cx) = c^{p-1} \nabla g(cx).$$

Therefore, the maximum is again attained at the boundary of the sub-box. The elementwise maximum is given by the roots of one-dimensional polynomials on fixed intervals and can be determined easily. This allows us to design the consistent convex underestimator selection τ_G of F on D. With two possible consistent convex underestimator selections at hand, we may apply the sufficient criteria for our separation strategy to work.

Similar observations as above also hold for $f(x) := x|x|$. The gradients are given by

$$\begin{aligned}
\nabla f(x) &= \begin{pmatrix} 2|x| \\ 0 \end{pmatrix}, \\
\nabla f(y) &= \begin{pmatrix} 0 \\ 2|y| \end{pmatrix}, \\
\nabla f(x+y) &= \begin{pmatrix} 2|x+y| \\ 2|x+y| \end{pmatrix}.
\end{aligned}$$

If the subdivision D_j of D is designed correctly, then the maximum of these gradients is again attained at the boundary of D_j. For this, let D_j be a sub-box and let the relative interior of each facet of D_j not contain points (x, y) with either $x = 0$, $y = 0$ or $x = y$ for all $j = 1, \dots, l$. Note that such a subdivision is always available, as motivated by the following example.

Example 3.40. Consider the function

$$g(x) = x|x| + \frac{1}{4}(x+y)^2 - y^2$$

on the box $D = [-6, 4] \times [1.5, 7.5]$. In order to derive a subdivision of D with the properties demanded above, we consider each orthant individually. For the second and forth orthant, every sub-box on the diagonal $x = y$ needs to have the same coordinates with respect to x and y. In this case, the discretization consists of 96 points and the subdivision of 77 sub-boxes (see Figure 3.5).

Note that g only consists of one absolute value term in this example. The requirements on the subdivision of D are therefore not necessary. However, we still use this example as an illustration because the convex envelope is available for g (see Chapter 4) and can be compared to the adjusted discretization.

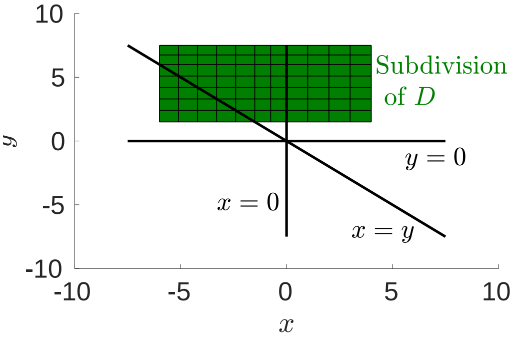


Figure 3.5: Example 3.40: Subdivision of D.

We derive the maximal gradient elementwise for every sub-box as discussed above. The resulting adjusted discretization $\tau[g]$ is illustrated in Figure 3.6, together with g and $\mathrm{vex}_D[g]$.

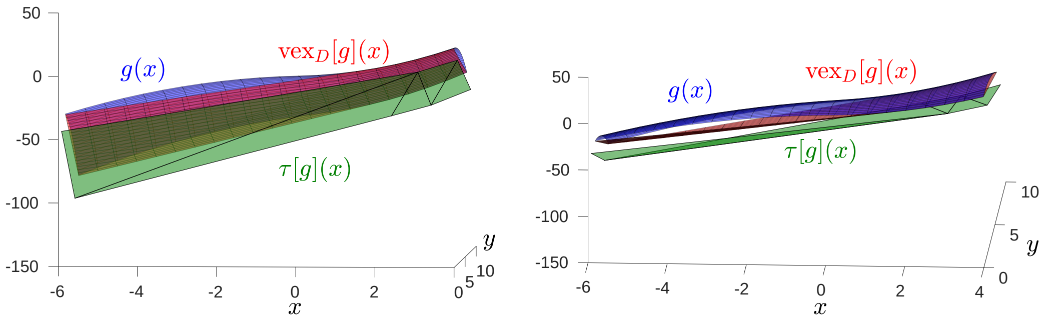


Figure 3.6: Example 3.40: Visualization of $g, \mathrm{vex}_D[g]$ and $\tau[g]$ from two different angles.

This type of quadratic absolute value functions arises from network design problems and is used in Chapter 4. We substitute the convex envelope by the adjusted discretization for the design of cutting planes. The effectiveness of the cutting planes is exemplarily evaluated on a test network.

For algorithmic purposes, it is important to note that we are only able to compute $\rho_G[f_\alpha](x)$ and $\tau_G[f_\alpha](x)$ for a given $\alpha \in B^m$ and $x \in D$ (see (3.7)).

This means that the adjusted discretization may only be used as a black box in order to solve the Approximate Problem 3.30. However, obvious solution strategies are subgradient methods that only rely on function values and subgradients for the iteration points. They can be applied in this setting, as subgradients can also be computed easily (see Corollary 3.19).

Part II

Problem Specific Optimization Techniques

Chapter 4

Convex Envelope and Cutting Planes for Gas Network Constraints

In this chapter we apply the general separation strategy for MINLPs that was developed in Chapter 3. The proposed strategy relies on an algorithmically utilizable representation of the convex envelope, that has to be available for every linear combination of the constraint functions. As deriving the convex envelope of arbitrary functions is beyond the capability of the current state of research, it is common to only consider a specific function class. In the following, we exemplarily restrict ourselves to bivariate quadratic absolute value functions. We derive the convex envelope of these kind of functions, which allows us to apply the separation strategy on the corresponding constraint sets. In order to derive the convex envelopes, we make use of Section 3.2 and various structural results from the literature (Tawarmalani and Sahinidis [2001]; Meyer and Floudas [2005]; Jach et al. [2008]).

The considered quadratic absolute value functions are used in the challenging field of gas network operation. We refer to [Koch et al., 2015] for an extensive introduction to this topic. A gas network in its simplest form is a system of connected pipes that is used to transport and distribute gas. The feasible set at every junction in this network can be modeled as the graph of a vector-valued function (see Problem 3.11). This allows us to evaluate the impact of our separation strategy on a real world application. Note that the considered functions are already studied in the literature, but with a different focus than provided in this work. See for example [Geißler et al., 2012; Pfetsch et al., 2015] for an approximation approach based on mixed-integer linear methods.

This chapter is structured as follows. In Section 4.1, we briefly describe the gas network setting and the resulting constraint structure. In particular, we focus on the description of a single junction in such a network. In Section 4.2, we present some analytic tools and concepts that help deriving the convex envelope for general function classes. We further exploit these results to derive the convex envelope of all linear combinations of relevant constraint functions for a single junction in a gas network. This allows us to apply the separation strategy from Section 3.2 to the respective feasible set. The practical impact of our work is evaluated for some small test instances of gas network optimization problems in Section 4.3. We derive stronger lower bounds using our separation strategy compared to the "standard" relaxation. Furthermore, we substitute the convex envelope in our separation problem by an estimation (see Section 3.3) and analyze the influence on the separation process.

This chapter is based on collaboration with Frauke Liers, Alexander Martin, Maximilian Merkert and Dennis Michaels. The author's contribution is presented in Sections 4.2 – 4.3. Preliminary considerations and computations are already published in [Merkert, 2017].

4.1 Constraint Structure in Gas Networks

A gas network in its simplest form consists of a system of connected pipes. In our setting, we neglect all other components like compressor stations or control valves. We consider the stationary case without dependency on time. Gas flows through the pipes based on the pressure differences at the respective end points. Mathematically, the network is modeled as a graph with arcs representing pipes and nodes representing end points. Most nodes only function as coupling points, which means that the difference of outgoing and incoming flow from all pipes is zero. For source nodes, this difference is positive, as the node is used to feed gas into the network. Sink nodes represent the demand of gas in

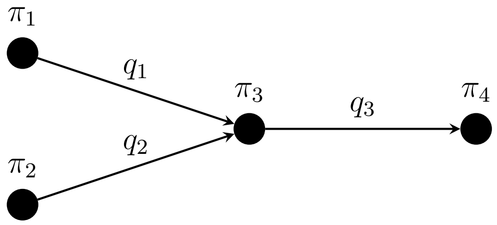


Figure 4.1: A single junction in a gas network.

the network, which means that the difference of outgoing and incoming flow is negative.

We focus on analyzing a single junction in a gas network consisting of four nodes and three arcs. Every arc has a corresponding flow variable q_j $(j = 1, 2, 3)$ and every node has a variable π_i $(i = 1, 2, 3, 4)$ that denotes the squared pressure. See Figure 4.1 for a visualization. We assume that the interior node 3 is neither a source nor a sink. The relevant constraints connecting these values are given by

$$\begin{aligned} c_1 \cdot |q_1|q_1 &= \pi_1 - \pi_3, \\ c_2 \cdot |q_2|q_2 &= \pi_2 - \pi_3, \\ c_3 \cdot |q_3|q_3 &= \pi_3 - \pi_4, \\ q_3 &= q_1 + q_2 \end{aligned} \tag{4.1}$$

with parameters $c \in \mathbb{R}^3$ (see Koch et al. [2015]).

Note that the direction of flow is given by the sign of each flow variable. The respective flow is directed as shown in Figure 4.1 for a positive sign, and the other way around for a negative sign.

We reformulate (4.1) into

$$\begin{aligned} \bar{g}_1(q_1, q_2, \pi_3) &:= \pi_1 = c_1 \cdot |q_1|q_1 + \pi_3, \\ \bar{g}_2(q_1, q_2, \pi_3) &:= \pi_2 = c_2 \cdot |q_2|q_2 + \pi_3, \\ \bar{g}_3(q_1, q_2, \pi_3) &:= \pi_4 = -c_3 \cdot |q_1 + q_2|(q_1 + q_2) + \pi_3. \end{aligned} \tag{4.2}$$

Function $\bar{g}$ is separable and π_3 is simply a linear term that can therefore be ignored. Parameter c is a scaling factor that has no influence on the structure of the remaining function. Furthermore, we identify $x_1 = q_1$, $x_2 = q_2$ and reduce our analysis to the feasible set

$$X := \big\{(x, z) \in \mathbb{R}^5 \mid z = g(x),\ x \in D\big\}$$

with $D \subseteq \mathbb{R}^2$ and

$$\begin{aligned} g_1(x) &:= |x_1|x_1, \\ g_2(x) &:= |x_1 + x_2|(x_1 + x_2), \\ g_3(x) &:= |x_2|x_2. \end{aligned} \tag{4.3}$$

The convex hull of the feasible set is again denoted by

$$Y := \operatorname{conv}(X).$$

Note that the complexity of the object X is mainly due to the absolute value terms in g. It is an obvious solution approach to branch all involved variables at 0 (see Section 2.1.3), and to consider the resulting subproblems without absolute values terms (see below). However, this results in exponentially many branches and subproblems, and is therefore not desirable.

Instead, we consider Y in its entirety and make use of the results from the previous chapter. For this, let a point $(\bar{x}, \bar{z}) \in D \times \mathbb{R}^3$ be given. In order to apply the proposed separation strategy from Section 3.2 on Y, we need to solve the Separation Problem 3.15 given as

$$\min_{\alpha \in B^3} h(\alpha) = \alpha^\top \bar{z} - \operatorname{vex}_D[\alpha^\top g](\bar{x}).$$

For this, we first derive the convex envelope of

$$g_\alpha := \alpha^\top g.$$

4.2 Convex Envelope of Quadratic Absolute Value Functions

In this section, we derive the convex envelope of g_α on $D \subseteq \mathbb{R}^2$ for arbitrary $\alpha \in B^3$ in order to solve Separation Problem 3.15. This is a challenging task in the general case. We reduce the complexity by assuming box constraints, i.e., $D = [l_1, u_1] \times [l_2, u_2]$, and by fixing the direction of flow in the considered junction.

When all flow directions are fixed, we can assume that the variables x_1 and x_2 are non-negative. This implies that the underlying functions reduce to quadratic ones and we obtain

$$Y = \operatorname{conv}\left(\{(x, x_1^2, x_2^2, (x_1 + x_2)^2) \mid x \in [l_1, u_1] \times [l_2, u_2]\}\right).$$

In this case, a complete description of Y is given in [Anstreicher and Burer, 2010].

The first case not covered by the literature is therefore given by two fixed flow directions and one variable flow direction. Without loss of generality, the only variable with unfixed sign is x_1. We assume that $x_2 \geq 0$ and $x_1 + x_2 \geq 0$ hold, so the single terms reduce to $g_1(x) = |x_1|x_1$, $g_2(x) = (x_1 + x_2)^2$ and $g_3(x) = x_2^2$. Thus, we are interested in determining the convex envelope of

$$g_\alpha(x) = \alpha_1 |x_1|\, x_1 + \alpha_2 (x_1 + x_2)^2 + \alpha_3 x_2^2$$

on $D = [l_1, u_1] \times [l_2, u_2]$ with $l_2 \geq 0$ for arbitrary $\alpha \in \mathbb{R}^3$. Function g_α is twice continuously differentiable for $x_1 \neq 0$. The Hessian matrix of g_α depends on the sign of x_1 and is given by

$$H_\alpha = \begin{cases} H_\alpha^-, & \text{if } x_1 < 0, \\ H_\alpha^+, & \text{if } x_1 > 0 \end{cases}$$

with

$$H_\alpha^- = 2 \begin{bmatrix} -\alpha_1 + \alpha_2 & \alpha_2 \\ \alpha_2 & \alpha_2 + \alpha_3 \end{bmatrix} \quad \text{and} \quad H_\alpha^+ = 2 \begin{bmatrix} \alpha_1 + \alpha_2 & \alpha_2 \\ \alpha_2 & \alpha_2 + \alpha_3 \end{bmatrix}.$$

Due to scaling we can assume $\alpha_1 \in \{-1, 0, 1\}$. In case of $\alpha_1 = 0$, g_α reduces to a quadratic function again. For the remainder of this section we restrict ourselves to $\alpha_1 = 1$, as the case $\alpha_1 = -1$ is similar and can be obtained by symmetric considerations.

There are nine remaining cases that need to be discussed and that depend on the specific values of the parameters α_2 and α_3. They define the curvature properties of g_α. In order to distinguish the different cases, we use the following definition.

Definition 4.1. Let $g : \mathbb{R}^n \to \mathbb{R}$ continuous and $D \subseteq \mathbb{R}^n$ convex.

- We call g direction-wise (strictly) convex/concave w.r.t. component i on D if, for every fixed $\bar{x} \in D$, g is (strictly) convex/concave on

$$\bar{D}(\bar{x}, i) := \{x \in D \mid x_j = \bar{x}_j \ \forall\, j = 1, \ldots, n,\ j \neq i\}.$$

- We call g indefinite on D if $\delta[g, \bar{x}] \neq \emptyset$ and $\xi[g, \bar{x}] \neq \emptyset$ hold for every $\bar{x} \in \operatorname{int}(D)$.

Case	Conditions	Curvature	Curvature	General
		w.r.t comp. 1	w.r.t comp. 2	curvature
1.	$-\alpha_2 \geq 1$	concave		
1.a	$\alpha_2 + \alpha_3 \leq 0$	concave	concave	concave/indefinite
1.b	$\alpha_2 + \alpha_3 > 0$	concave	convex	indefinite
2.	$\alpha_2 \geq 1$	convex		
2.a	$\alpha_2 + \alpha_3 \leq 0$	convex	concave	indefinite
2.b.	$\alpha_2 + \alpha_3 > 0$	convex	convex	
2.b.i.	$H_\alpha^+ \not\succcurlyeq 0$	convex	convex	indefinite
2.b.ii.	$H_\alpha^- \succcurlyeq 0$	convex	convex	convex
2.b.iii.	$H_\alpha^+ \succcurlyeq 0$, $H_\alpha^- \not\succcurlyeq 0$	convex	convex	indefinite-convex
3.	$-1 < \alpha_2 < 1$	concave-convex		
3.a	$\alpha_2 + \alpha_3 < 0$	concave-convex	concave	concave/indefinite
3.b.	$\alpha_2 + \alpha_3 \geq 0$	concave-convex	convex	
3.b.i	$H_\alpha^+ \not\succcurlyeq 0$	concave-convex	convex	indefinite
3.b.ii	$H_\alpha^+ \succcurlyeq 0$	concave-convex	convex	indefinite-convex

Table 4.1: Conditions and properties for all nine (sub)cases.

Using this notation, the nine different cases can be distinguished as listed in Table 4.2. The first column denotes the (sub-)cases and the second one lists the conditions on α for the respective case. Columns three to five give the curvature of g_α with respect to both components and in general. *Concave-convex* means that g_α is direction-wise concave for $x < 0$ and direction-wise convex for $x \geq 0$. *Indefinite-convex* means that g_α is indefinite for $x < 0$ and convex for $x \geq 0$. *Concave/indefinite* indicates that g_α is either concave or indefinite.

In the following, we first discuss some interesting results and properties of the convex envelope exemplarily on the two most complicated (sub-)cases. Section 4.2.1 considers Case (2.b.iii) and uses the concept of $(n-1)$-*convex functions* (see Jach et al. [2008]) in order to show that the convex envelope of g_α

consists of minimizing segments. In Section 4.2.2, we introduce the *direction-wise convex envelope* and reduce Case (3) to Case (2). The remaining cases are briefly discussed in Section 4.2.3. They are obtained by results in the literature or by similar arguments as presented.

Note that the following presentation is very extensive and in some cases quite complicated. However, the computational effort needed to make use of the results is very little. In fact, for most cases we obtain a minimizing simplex for every point $x \in D$ analytically. The exceptions are Case (2.b.i), (2.b.iii), (2.c.i) and (2.c.ii). For these cases we need the roots of a polynomial of degree four, that can also be computed with little effort. Furthermore, the minimizing simplices directly result in the objective value and a subgradient of Problem 3.15 for given $\alpha \in B^3$ and $(\bar{x}, \bar{z}) \in D \times \mathbb{R}^3$ (see Section 3.2.3).

4.2.1 Reduction on Minimizing Segments

We consider Case (2.b.iii). The function g_α is direction-wise convex w.r.t component 1 and 2. Furthermore, it is convex for $x_1 \geq 0$ and indefinite for $x_1 < 0$. We show that the convex envelope consists of minimizing segments and derive them for any given point $\bar{x} \in D$.

For this, we make use of the concept of $(n-1)$-convex functions as introduced in [Jach et al., 2008].

Definition 4.2. Let $g : \mathbb{R}^n \to \mathbb{R}$ be a twice differentiable function. g is said to be (strictly) $(n-1)$-convex if the function $g|_{x_i=\bar{x}_i} : \mathbb{R}^{n-1} \to \mathbb{R}$ is (strictly) convex for each fixed value $\bar{x}_i \in \mathbb{R}$ and for all $i = 1, \ldots, n$.

Remark 4.3. For $g : \mathbb{R}^n \to \mathbb{R}$ and $n = 2$, g being (strictly) $(n-1)$-convex is equivalent to g being direction-wise (strictly) convex w.r.t. both components.

For indefinite functions with this property, the authors make a statement on the structure of the concave directions.

Lemma 4.4. [Jach et al., 2008, Lemma 3.2] *Let $g : D = [l, u] \subseteq \mathbb{R}^n \to \mathbb{R}$ be a twice differentiable function, and let the collection $\{\mathcal{O}_1, \ldots, \mathcal{O}_{2^n}\}$ be the system of open orthants of the space $\mathbb{R}^n$. Then, the function g is $(n-1)$-convex and indefinite if and only if $\delta[g, x]$ is nonempty for each $x \in D$ and there exists an* *index $i \in \{1, \dots, 2^n\}$, such that*

$$\delta[g, x] \subseteq \mathcal{O}_i \cup (-\mathcal{O}_i)$$

holds for all $x \in D$.

This statement can be extended to the function g_α for Case (2.b.iii). g_α can be divided into an indefinite $(n-1)$-convex function for negative x_1 and a convex function for positive x_1. The property stated in Lemma 4.4 therefore also holds for g_α as shown in the following Corollary.

Corollary 4.5. *Let α be as given in Case (2.b.iii). Then, there exists an index $i \in \{1, \dots, 2^n\}$, such that*

$$\delta[g_\alpha, x] \subseteq \mathcal{O}_i \cup (-\mathcal{O}_i)$$

holds for all $x \in D$.

Proof. We divide function g_α into two parts and formulate it as

$$g_\alpha(x) = \begin{cases} g_\alpha^-(x), & \text{if } x_1 < 0, \\ g_\alpha^+(x), & \text{if } x_1 \geq 0 \end{cases}$$

with

$$\begin{aligned} g_\alpha^-(x) &:= -\alpha_1 x_1^2 + \alpha_2 (x_1 + x_2)^2 + \alpha_3 x_2^2 \\ \text{and} \quad g_\alpha^+(x) &:= \alpha_1 x_1^2 + \alpha_2 (x_1 + x_2)^2 + \alpha_3 x_2^2. \end{aligned}$$

The function g_α^+ is convex because of $H_\alpha^+ \succcurlyeq 0$, so we have

$$\delta[g_\alpha, x] = \begin{cases} \delta[g_\alpha^-, x], & \text{if } x_1 < 0, \\ \emptyset, & \text{if } x_1 > 0. \end{cases}$$

The concave directions at a point x with $x_1 = 0$ need to be discussed separately. We consider the direction $d = (d_1, d_2)$ and distinguish the two cases $d_1 = 0$ and $d_1 \neq 0$. For $d_1 = 0$, the function

$$h_{g,x,d}(\lambda) = g_\alpha(x + \lambda d) \qquad \text{(see Definition 3.3)}$$

is convex in λ as g_α is direction-wise convex w.r.t. component 2. Therefore, any d with $d_1 = 0$ is not a concave direction. For $d_1 \neq 0$, the function $h_{g,x,d}(\lambda)$ with

$\lambda \in [-\varepsilon, \varepsilon]$ always has a domain that includes points whose first component attains values strictly greater zero for every $\varepsilon > 0$. As g_α is convex for $x_1 > 0$, d is again not a concave direction.

Summarizing these results, we obtain

$$\delta[g_\alpha, x] = \begin{cases} \delta[g_\alpha^-, x], & \text{if } x_1 < 0, \\ \emptyset, & \text{if } x_1 \geq 0 \end{cases}$$

for every $x \in D$. As g_α^- is an $(n-1)$-convex and indefinite function, we can apply Lemma 4.4 to conclude that there exists an index $i \in \{1, 2, 3, 4\}$, such that for all $x \in D$ the set of concave direction of g_α at x is a subset of $\mathcal{O}_i \cup -\mathcal{O}_i$. □

This structure of the concave directions can be used to show the existence of minimizing segments for every point $\bar{x} \in D$ w.r.t. a set G (see Definition 3.5). As the existence of minimizing segments depends on the choice of G, we first define $G := G_1 \cup G_2 \cup G_3 \cup G_4$ with

$$\begin{aligned} G_1 &:= \{l_1\} \times [l_2, u_2], & G_2 &:= [l_1, 0] \times \{l_2\}, \\ G_3 &:= [l_1, 0] \times \{u_1\}, & G_4 &:= [0, u_1] \times [l_2, u_2]. \end{aligned}$$

See Figure 4.2(a) for a graphic representation of the subsets G_i. Using Observation 3.4, it is easy to see that $\mathfrak{G}[g_\alpha, D] \subseteq G$ holds.

The existence of minimizing segments is then given by the following Lemma. A similar case and basic ideas of the proof are already provided in [Jach et al., 2008, Theorem 3.1].

Corollary 4.6. *Let α be as given in Case (2.b.iii), $D = [l, u]$ with $l_2 \geq 0$ and G as defined above. Then the convex envelope of g_α on D consists of minimizing segments w.r.t. G.*

Proof. Note the following preliminary considerations. Let

$$E := D \setminus G = \big\{p \in \operatorname{int}(D) \mid p_1 < 0\big\}.$$

The proof of Corollary 4.5 already indicates that g_α^- is indefinite. The set of concave directions of g_α at $x \in E$ is therefore non-empty.

For the proof, we mainly show that the convex envelope of g on D consists of minimizing segments w.r.t. D. Using Observation 3.8, it is easy to see that there are no extreme points of any minimizing segment inside E. We conclude that the convex envelope of g on D also consists of minimizing segments w.r.t. G.

Assume that there exists a point $\bar{x} \in D$ with a minimizing simplex $\mathcal{S}_{g_\alpha,D}(\bar{x})$ consisting of at least three different extreme points, i.e.,

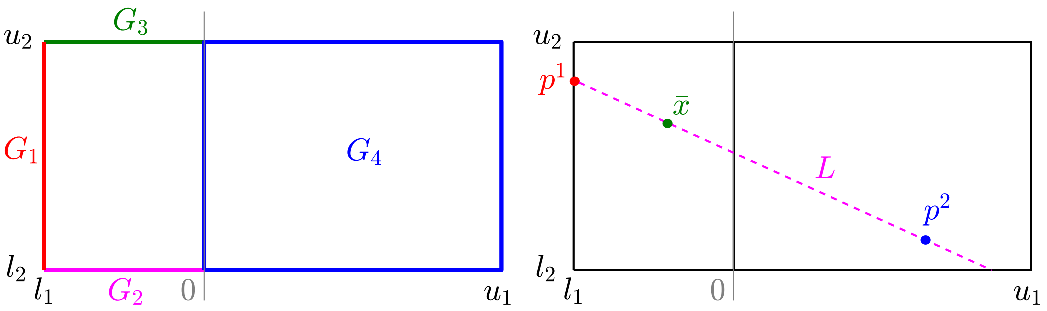
$$x^1, x^2, x^3 \in \operatorname{extr}\big(\mathcal{S}_{g_\alpha,D}(\bar{x})\big) \quad \text{with} \quad x^1 \neq x^2 \neq x^3 \neq x^1.$$

According to Observation 3.8, we have $x^1, x^2, x^3 \notin E$, as $\delta[g_\alpha, p] \neq \emptyset$ holds for all points $p \in E$. We conclude that $x^1, x^2, x^3 \in G$ and we only need to distinguish two cases:

- Two of the points x^1, x^2, x^3 are elements of the same subset G_i for some $i \in \{1, \ldots, 4\}$. Function g_α is convex on G_i for every $i \in \{1, \ldots, 4\}$. This leads to a contradiction based on Observation 3.8.

- No two points are elements of the same subset G_i for all $i \in \{1, \ldots, 4\}$. For all possible combinations, one of the three vectors $(x^1 - x^2)$, $(x^2 - x^3)$ and $(x^3 - x^1)$ is not element of the same pair of open orthants $\mathcal{O}_i \cup (-\mathcal{O}_i)$ for all $i = 1, \ldots, 4$. According to Corollary 4.5, g_α is convex on at least one of the three sets $\operatorname{conv}\big(\{x^1, x^2\}\big)$, $\operatorname{conv}\big(\{x^2, x^3\}\big)$ or $\operatorname{conv}\big(\{x^3, x^1\}\big)$. This contradicts Observation 3.8 again.

Hence, the convex envelope of g on D consists of minimizing segments w.r.t. D. We conclude our statement by using Observation 3.8 as described above. □



(a) Visualization of the subdivision of G.

(b) Visualization of L, p^1 and p^2.

Figure 4.2: Case (2.b.iii).

Next, we construct a minimizing segment for any given point $\bar{x} \in D$ w.r.t g_α and G. For this, we first analyze the structure of concave directions in order to apply Observation 3.8. For $\alpha_2 > 1$ we show

$$(-\alpha_2, \alpha_2 - \alpha_1) \in \delta[g_\alpha, x]$$

for all $x \in D$ with $x_1 < 0$. In fact, we have

$$\begin{aligned} &(-\alpha_2, \alpha_2 - \alpha_1)\, H_\alpha^- (-\alpha_2, \alpha_2 - \alpha_1)^\top \\ &= \alpha_2^2(\alpha_2 - \alpha_1) - 2\alpha_2^2(\alpha_2 - \alpha_1) + (\alpha_2 - \alpha_1)^2(\alpha_2 + \alpha_3) \\ &= (\alpha_2 - \alpha_1)\det(H_\alpha^-). \end{aligned}$$

With $\alpha_2 > 1 = \alpha_1$ and $H_\alpha^- \not\succcurlyeq 0$, we obtain $\det(H_\alpha^-) < 0$ and

$$(-\alpha_2, \alpha_2 - \alpha_1) \in \delta[g_\alpha, x].$$

For $\alpha_2 = 1$, it is easy to see that

$$(-\alpha_2 + \alpha_3, 1)\, H_\alpha^- (-\alpha_2 + \alpha_3, 1)^\top < 0$$

holds for all $x \in D$ with $x_1 < 0$. This leads to

$$(-\alpha_2 + \alpha_3, 1) \in \delta[g_\alpha, x].$$

Either way, there exists a vector $v \in \mathbb{R}_- \times \mathbb{R}_+$ with $v \in \delta[g_\alpha, x]$. Using Corollary 4.5, we derive

$$\delta[g_\alpha, x] \subseteq \operatorname{int}(\mathbb{R}_+ \times \mathbb{R}_-) \cup \operatorname{int}(\mathbb{R}_- \times \mathbb{R}_+) \tag{4.4}$$

for all $x \in D$. This property of the structure of concave directions will be used together with Observation 3.8 in the following analysis.

By Corollary 4.6, there exists a minimizing segment $\mathcal{S}_{g_\alpha,G}(\bar{x})$ for any given point $\bar{x} \in D$. We denote the two extreme points of $\mathcal{S}_{g_\alpha,G}(\bar{x})$ by p^1 and p^2, i.e.,

$$\mathcal{S}_{g_\alpha,G}(\bar{x}) = \operatorname{conv}(p^1, p^2).$$

By definition of G, we have $p^1 \in G_i$ and $p^2 \in G_j$ for some $i, j \in \{1, \ldots, 4\}$. Next, we classify possible minimizing segments for all combinations of i and j (exploiting symmetry). For this, consider the following "easy" cases first.

$i = j$: As g_α is convex on G_k for $k = 1, \dots, 4$, we have $p^1 = p^2 = \bar{x}$ in this case.

$(i, j) = (1, 3)$: Using (4.4) and Observation 3.8, we derive that there are no minimizing segments with $p^1 \in G_1$ and $p^2 \in G_3$ except for $p^1 = p^2 = \bar{x} = (l_1, u_2)$.

$(i, j) = (2, 4)$: Using (4.4) and Observation 3.8 again, this leads to $p^1 = p^2 = \bar{x} = (0, l_2)$.

$(i, j) = (2, 3)$: See Case (2.a) in Section 4.2.3.

$(i, j) = (1, 2)$: See Case (2.b.i) in Section 4.2.3.

The first interesting combination is $(i, j) = (1, 4)$. For every given point $\bar{x} \in D$ and every extreme point $p^1 := (l_1, r) \in G_1$ of a possible minimizing segment of $\bar{x}$, we consider the ray L starting at p^1 into the direction of $\bar{x}$ as

$$L := \left\{ p^1 + \lambda(\bar{x} - p^1) \mid \lambda \geq 0 \right\}.$$

We determine the convex envelope of g_α restricted to L, and thereby detect the second extreme point $p^2 := (s, t) \in G_4$ (See Figure 4.2(b)). The point p^2 is given as the point with a directional derivative coinciding with the gradient of the line connecting $\bigl(p^1, g_\alpha(p^1)\bigr)$ and $\bigl(p^2, g_\alpha(p^2)\bigr)$, i.e.,

$$g_\alpha(p^2) + \nabla g_\alpha(p^2)^\top (p^1 - p^2) = g_\alpha(p^1).$$

Note that $s \geq 0$ holds because of $p^2 \in G_4$. We further introduce a new variable μ and set

$$p^2 = p^1 + \mu(\bar{x} - p^1) \quad \Leftrightarrow \quad \begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} l_1 \\ r \end{pmatrix} + \mu \begin{pmatrix} \bar{x}_1 - l_1 \\ \bar{x}_2 - r \end{pmatrix}.$$

Variable μ can be interpreted as the distance between p^1 and p^2 relatively to the distance between p^1 and $\bar{x}$. We combine the equations above and derive

$$\mu = \frac{-\sqrt{2\alpha_1}\, l_1}{\sqrt{(\alpha_1 + \alpha_2)(\bar{x}_1 - l_1)^2 + 2\alpha_2(\bar{x}_1 - l_1)(\bar{x}_2 - r) + (\alpha_2 + \alpha_3)(\bar{x}_2 - r)^2}}.$$

Each of the variables r, s, t and μ now depend on r. We insert this information into the problem used to derive the value of the convex envelope at a given point $\bar{x}$ (see (3.1)). This results in the one-dimensional optimization problem

$$\begin{aligned} \min\ h(r) := {} & \frac{1}{\mu}\, g_\alpha(s,t) + \left(1 - \frac{1}{\mu}\right) g_\alpha(l_1, r) \\ \text{s.t.}\ \ & s = l_1 + \mu(\bar{x}_1 - l_1) \geq 0 \\ & t = r + \mu(\bar{x}_2 - r) \\ & \mu = \tfrac{-\sqrt{2\alpha_1}\, l_1}{\sqrt{(\alpha_1+\alpha_2)(\bar{x}_1-l_1)^2+2\alpha_2(\bar{x}_1-l_1)(\bar{x}_2-r)+(\alpha_2+\alpha_3)(\bar{x}_2-r)^2}} \\ & r \in [l_2, u_2]. \end{aligned} \tag{4.5}$$

This problem has to be solved in order to determine the actual minimizing segment and the value of the convex envelope at $\bar{x}$. Using basic transformation, we first reformulate the objective function into

$$h(r) = r\Big(2\alpha_2(\bar{x}_1 - l_1) + 2(\alpha_2 + \alpha_3)\bar{x}_2\Big) + r^2(\alpha_2 + \alpha_3) + \frac{1}{\mu}(4\alpha_1 l_1^2) + c$$

with a constant c not depending on r, that can be omitted for sake of optimization. We aim to apply the first order optimality condition and consider the derivative of $h(r)$ (with μ also depending on r), which reads as

$$h'(r) = \Big(2\alpha_2(\bar{x}_1 - l_1) + 2(\alpha_2 + \alpha_3)(\bar{x}_2 - r)\Big)\Big(1 - \mu\Big).$$

In order to determine the optimal solution, the roots of $1 - \mu = 0$ do not have to be considered as this would result in $p^2 = \bar{x}$. The remaining root of the derivative is given by

$$r_1 = \bar{x}_2 + \frac{\alpha_2}{\alpha_2 + \alpha_3}(\bar{x}_1 - l_1).$$

Hence, the optimal value of (4.5) is attained at r_1 or at the boundary of the interval $[l_2, u_2]$. The minimum of (4.5) can not be attained at $r = l_2$, because this would result in a minimizing segment not including a concave direction (see (4.4) and Observation 3.8). The two remaining possible optimal solutions are therefore $r_1 = \bar{x}_2 + \frac{\alpha_2}{\alpha_2+\alpha_3}(\bar{x}_1 - l_1)$ and $r_2 = u_2$, and their respective minimizing segments.

For $(i, j) = (3, 4)$, the resulting possible minimizing segment can be derived in a simultaneous way. Combining these results, we derive possible minimizing segments for several combinations of $i, j \in \{1, \ldots, 4\}$. As all combinations are considered, the actual minimizing segment for $\bar{x}$ has to be one of them. In order to determine it, we compute the values of the convex combination induced by all different possible segments and take the lowest one.

Figure 4.3 exemplarily shows the resulting structure of the minimizing segments for different $\bar{x} \in D$. We use green lines for segments with $(i, j) = (1, 2)$, magenta and red lines for segments with $(i, j) = (1, 4)$ and blue lines for segments with $(i, j) = (3, 4)$. Yellow lines show intermediate segments with one extreme point at (l_1, u_2). Black dots indicate minimizing segments of dimension zero inside set G_4. This means that the function g_α coincides with its convex envelope.

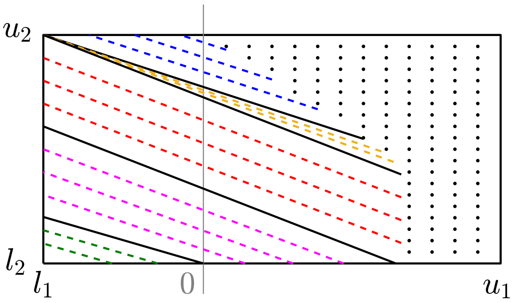


Figure 4.3: Case (2.b.iii): Structure of the minimizing segments.

4.2.2 The Direction-Wise Convex Envelope

We consider Case (3) and all its sub-cases. In order to handle these cases, we introduce the concept of direction-wise convex envelopes and show how it can be used to reduce Case (3) to results from Case (2).

$\alpha_2 \in (-1, 1)$ holds, so that, w.r.t. component 1, g_α is direction-wise concave for $x \leq 0$ and direction-wise convex for $x \geq 0$. For a function g that is not direction-wise convex w.r.t to a certain coordinate $i \in \{1, \ldots, n\}$ on the whole set D, we can design a function with this property by computing the convex envelope of g restricted to a line segment defined by fixing the value of x_j for all $j \neq i$, $i \in \{1, \ldots, n\}$.

Definition 4.7. The direction-wise convex envelope of g on D w.r.t. component i is defined as

$$\begin{aligned} &\gamma_{D,i}[g](x) := \operatorname{vex}_{\bar{D}(x,i)}[g](x) \\ \text{with} \qquad &\bar{D}(x,i) = \{x' \in D \mid x'_j = x_j \; \forall j = 1, \ldots, n \; j \neq i\}. \end{aligned} \tag{4.6}$$

In certain cases, this operation preserves the direction-wise curvature with respect to other coordinates.

Lemma 4.8. *Let $g : \mathbb{R}^n \to \mathbb{R}$ continuous and direction-wise (strictly) convex/concave w.r.t component $k \in \{1, \ldots n\}$. Let $D \subseteq \mathbb{R}^n$ be a closed convex set and let $i \in \{1, \ldots, n\}$ be given. If there exists a set $X_i \subset \mathbb{R}$ such that*

$$\mathfrak{G}[g, \bar{D}(x, i)] = \big\{y \in D \mid y_i \in X_i,\ y_j = x_j \ \forall\ j = 1, \ldots, n,\ j \neq i\big\}$$

holds for every $x \in D$, then $\gamma_{D,i}[g]$ is direction-wise (strictly) convex/concave w.r.t. component k.

Proof. We discuss the proof only for the statement on convexity. The results for concavity and strictness can be derived analogously.

Let $\lambda \in [0, 1]$ and two points x^1, x^2 with $x^1_j = x^2_j$ for every $j = 1, \ldots, n,\ j \neq k$ be given. We denote $x^\lambda := \lambda x^1 + (1 - \lambda)x^2$. Due to our condition, we have either

$$x^1 \in \mathfrak{G}[g, \bar{D}(x^1, i)], \quad x^2 \in \mathfrak{G}[g, \bar{D}(x^2, i)], \quad x^\lambda \in \mathfrak{G}[g, \bar{D}(x^\lambda, i)]$$

or

$$x^1 \notin \mathfrak{G}[g, \bar{D}(x^1, i)], \quad x^2 \notin \mathfrak{G}[g, \bar{D}(x^2, i)], \quad x^\lambda \notin \mathfrak{G}[g, \bar{D}(x^\lambda, i)].$$

In the first case, $\gamma_{D,i}[g](x) = g(x)$ holds for all $x \in \{x^1, x^2, x^\lambda\}$ and

$$\lambda \gamma_{D,i}[g](x^1) + (1 - \lambda)\gamma_{D,i}[g](x^2) \geq \gamma_{D,i}[g](x^\lambda) \tag{4.7}$$

holds as g is direction-wise convex w.r.t. component k.

In the second case, the value of $\gamma_{D,i}[g](x)$ for $x \in \{x^1, x^2, x^\lambda\}$ is given as a convex combination of two points respectively. This holds as the direction-wise convex envelope is defined on a one-dimensional set and therefore always consists of minimizing segments. Due to (4.7), the respective two points share the same value for every component but component k for all $x \in \{x^1, x^2, x^\lambda\}$. To be more specific, there exists some $\mu \in [0, 1]$, and points

$$x^{1,1}, x^{1,2}, x^{2,1}, x^{2,2}, x^{\lambda,1}, x^{\lambda,2} \in D$$

with $x_j^{1,1} = x_j^{2,1} = x_j^{\lambda,1}$ and $x_j^{1,2} = x_j^{2,2} = x_j^{\lambda,2}$ for every $j = 1, \dots, n,\ j \neq k$, such that

$$\begin{aligned}\gamma_{D,i}[g](x^1) &= \mu g(x^{1,1}) + (1-\mu) g(x^{1,2}),\\ \gamma_{D,i}[g](x^2) &= \mu g(x^{2,1}) + (1-\mu) g(x^{2,2}),\\ \gamma_{D,i}[g](x^\lambda) &= \mu g(x^{\lambda,1}) + (1-\mu) g(x^{\lambda,2}).\end{aligned}$$

As $\lambda g(x^{1,1}) + (1-\lambda) g(x^{2,1}) \geq g(x^{\lambda,1})$ and $\lambda g(x^{1,2}) + (1-\lambda) g(x^{2,2}) \geq g(x^{\lambda,2})$ holds due to the direction-wise convexity of g, we again obtain

$$\lambda \gamma_{D,i}[g](x^1) + (1-\lambda) \gamma_{D,i}[g](x^2) \geq \gamma_{D,i}[g](x^\lambda).$$

□

Furthermore, the direction-wise convex envelope is a suitable intermediate step for determining the actual convex envelope.

Corollary 4.9. *Let $D \subseteq \mathbb{R}^n$ and $g : D \to \mathbb{R}$ continuous. Then*

$$\mathrm{vex}_D[g](x) = \mathrm{vex}_D[\gamma_{D,i}[g]](x).$$

holds for all $i \in \{1, \dots, n\}$.

In order to make use of this result, we first derive the direction-wise convex envelope of g_α w.r.t. component 1. It is

$$\gamma_{D,1}[g_\alpha](x) = \begin{cases} g_\alpha(l_1, x_2) + (x_1 - l_1) \frac{g_\alpha(s, x_2) - g_\alpha(l_1, x_2)}{s - l_1}, & \text{if } x_1 \leq s,\\ g_\alpha(x), & \text{if } x_1 > s \end{cases}$$
$$\text{with} \quad s := \min\left(u_1,\ l_1\left(1 - \sqrt{1 - \tfrac{\alpha_2 - 1}{\alpha_2 + 1}}\right)\right).$$

See the following example for a small illustration.

Example 4.10. We consider $\alpha = (1, -0.25, 1)$ and $D = [-1, 1] \times [0, 1]$. Figure 4.4 shows the direction-wise convex envelope of g_α w.r.t component 1 for $x_2 = 0$ and $x_2 = 1$ respectively. Note that the extreme points of the minimizing segment $[-1, s]$ are the same in both cases. This holds as the value of x_2 only changes the first derivative of g_α w.r.t. x_1, but not its overall curvature. Therefore, s is independent of x_2.

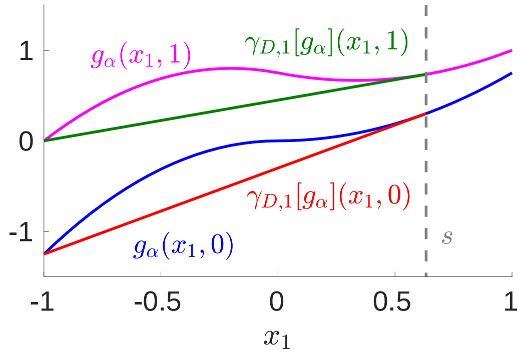


Figure 4.4: Illustration of Example 4.10: The direction-wise convex envelope for $x_2 = 0$ and $x_2 = 1$ respectively.

$\gamma_{D,1}[g_\alpha](x)$ restricted to $x_1 < s$ is twice differentiable, so we can derive the Hessian Matrix as

$$H_{\gamma_{D,1}[g_\alpha](x)|_{x_1<s}} = 2 \begin{bmatrix} 0 & \alpha_2(s-l_1) \\ \alpha_2(s-l_1) & (\alpha_2+\alpha_3) \end{bmatrix}.$$

$\gamma_{D,1}[g_\alpha]$ is direction-wise convex w.r.t. component 1 and indefinite for $x_1 < s$. Note that, for all sub-cases of Case (3), the direction-wise convexity/concavity w.r.t component 2 is also preserved as stated in Lemma 4.8. This holds as the value of s in the definition of $\gamma_{D,1}[g_\alpha]$ is independent in x_2. By applying these results, the convex envelope of $\gamma_{D,1}[g_\alpha]$ for all sub-cases of Case (3) can be reduced to observations in other cases. See Case (3.a), (3.b.i) and (3.b.ii) in Section 4.2.3 for detailed information.

Using the convex envelope of $\gamma_{D,1}[g_\alpha]$, the convex envelope of g_α can also be easily derived by translating the minimizing segments w.r.t. $\gamma_{D,x}[g_\alpha]$ into minimizing simplices w.r.t. g_α. For this, we consider any extreme point y of a minimizing segment w.r.t. $\gamma_{D,x}[g_\alpha]$. If $y_1 < s$ holds, then we derive two extreme points (l_1, y_2) and (s, y_2) of the respective minimizing simplex w.r.t. g_α. Otherwise the extreme point y is also an extreme point of the respective minimizing simplex w.r.t. g_α.

4.2.3 Summary of the Remaining Cases

The respective convex envelope and minimizing segments of the remaining cases are obtained by results from the literature or by similar considerations as given in Section 4.2.1. We briefly present them in the following for the sake of completeness. We also refer to the technical report [Ballerstein et al., 2013] for similar considerations.

Case (1.a)

In this case, g_α is direction-wise concave w.r.t components 1 and 2. As our domain D is a box, $g_\alpha(x)$ is also called *edge-concave*. Functions with this property and the respective convex envelopes are for example studied in [Meyer and Floudas, 2005].

The generating set of g_α is given by the four extreme points of the box, i.e.,

$$\mathfrak{G}[g_\alpha, D] = \big\{(l_1, l_2), (l_1, u_2), (u_1, l_2), (u_1, u_2)\big\}.$$

The convex envelope is polyhedral and the minimizing simplices are induced by a certain triangulation of the box D. As we deal with a bivariate function, D can be triangulated in only two different ways:

- Triangulation T_1 is given by the sets $G_1 := \big\{(l_1, u_2), (l_1, l_2), (u_1, u_2)\big\}$ and $G_2 := \big\{(l_1, l_2), (u_1, u_2), (u_1, l_2)\big\}$.
- Triangulation T_2 is given by the sets $G_3 := \big\{(l_1, l_2), (l_1, u_2), (u_1, l_2)\big\}$ and $G_4 := \big\{(l_1, u_2), (u_1, l_2), (u_1, u_2)\big\}$.

In order to decide which of the two possible triangulations determines the convex envelope, we need to compare the respective values at the center point $\frac{1}{2}(l_1+u_1, l_2+u_2)$ of D (e.g., see Meyer and Floudas [2005]). The corresponding possible minimizing segments for the center point are given by

$$\operatorname{conv}\big(\{(l_1, l_2), (u_1, u_2)\}\big) \quad \text{and} \quad \operatorname{conv}\big(\{(l_1, u_2), (u_1, l_2)\}\big).$$

It turns out that

$$\tfrac{1}{2}\big(g_\alpha(l_1, l_2) + g_\alpha(u_1, u_2)\big) \leq \tfrac{1}{2}\big(g_\alpha(l_1, u_2) + g_\alpha(u_1, l_2)\big)$$

holds for $x_2 \geq 0$ and that, hence, triangulation T_1 determines the convex envelope. The resulting minimizing simplices are given by $\operatorname{conv}(G_1)$ (green region in Figure 4.5), $\operatorname{conv}(G_2)$ (blue region in Figure 4.5) and $\operatorname{conv}\big(\{(l_1, l_2), (u_1, u_2)\}\big)$ (red line in Figure 4.5).

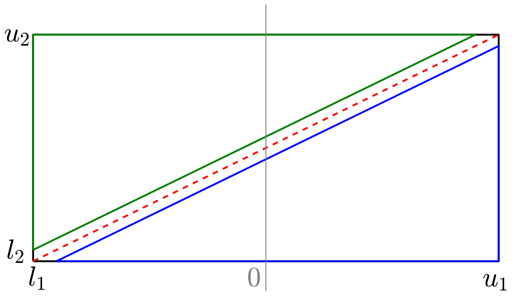


Figure 4.5: Case (1.a): Triangulation of D induced by T_1.

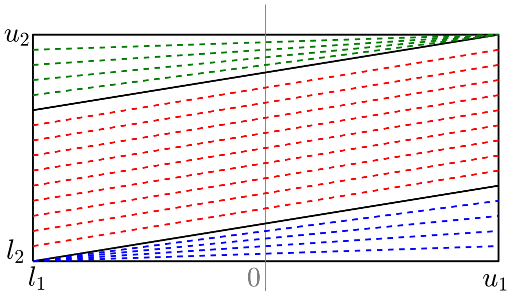


Figure 4.6: Case (1.b): Structure of the minimizing segments.

Case (1.b)

In this case, g_α is direction-wise concave w.r.t component 1 and direction-wise convex w.r.t. component 2. The generating set of g_α is given as

$$\mathfrak{G}[g_\alpha, D] = \{l_1, u_1\} \times [l_2, u_2].$$

As $g_\alpha(x)$ is convex on $\{l_1\}\times[l_2, u_2]$ and $\{u_1\}\times[l_2, u_2]$ respectively, no minimizing simplex contains more than one point in each of both subsets. Hence, the convex envelope w.r.t. $\mathfrak{G}[g_\alpha, D]$ consists of minimizing segments of the form $\operatorname{conv}\big((l_1, y_1), (u_1, y_2)\big)$, with $y_1, y_2 \in [l_2, u_2]$. Functions of this type are already studied in the literature (e.g., Tawarmalani and Sahinidis [2001]; Jach et al. [2008]).

For a given point $\bar{x}$, the specific values of y_1 and y_2 are given as the unique minimizer of an univariate optimization problem. They need to satisfy

$$\frac{\partial g_\alpha}{\partial x_2}(l_1, y_1) = \frac{\partial g_\alpha}{\partial x_2}(u_1, y_2) \tag{4.8}$$

or either y_1 or y_2 need to lie at the boundary of the interval $[l_2, u_2]$.

Thus, a minimizing segment is either parallel to the vector $v := \left(1, -\frac{\alpha_2}{\alpha_2+\alpha_3}\right)$ (red lines in Figure 4.6), or is determined by $y_1 = l_2$ (blue lines in Figure 4.6) or by $y_2 = u_2$ (green lines in Figure 4.6).

Case (2.a)

In this case, g_α is direction-wise convex w.r.t component 1 and direction-wise concave w.r.t. component 2. The generating set of g_α is given as

$$\mathfrak{G}[g_\alpha, D] = [l_1, u_1] \times \{l_2, u_2\}.$$

In order to compute the minimizing segments, we use the same arguments as in Case (1.b) with inverted roles of the two coordinates.

As the second derivative of g_α differs among the two half-spaces $x_1 \leq 0$ and $x_1 \geq 0$, we additionally distinguish three possibilities defined by the position of the minimizing segments with respect to these half-spaces. Minimizing segments containing only points with negative values of x_1 are parallel to the vector $v^1 := \left(\frac{\alpha_2}{\alpha 2 - \alpha_1}, -1\right)$ (yellow lines in Figure 4.7), or defined by the extreme point (l_1, u_2) (green lines in Figure 4.7). Segments containing only points with a positive value of x_1 are parallel to the vector $v^2 := \left(\frac{\alpha_2}{\alpha 2 + \alpha_1}, -1\right)$ (red lines in Figure 4.7) or defined by the extreme point (u_1, l_2) (blue lines in Figure 4.7). Minimizing segments containing points from both half spaces are not parallel to each other. For one extreme points (y_1, u_2), the second one is given by $\left(y_1 + \frac{\alpha_2(u_2 - l_2) - 2\alpha_1 y_1}{\alpha_1 + \alpha_2}, l_2\right)$ (magenta lines in Figure 4.7).

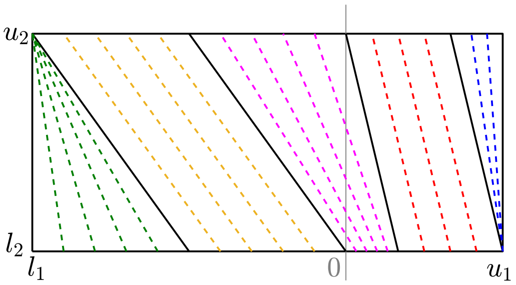


Figure 4.7: Case (2.a): Structure of the minimizing segments.

Case (2.b.i)

In this case, g_α is indefinite and direction-wise convex w.r.t. components 1 and 2. We derive a similar result as in Section 4.2.1 for Case (2.b.iii).

Corollary 4.11. *Let α be as defined in Case (2.b.i) and let $G = \mathrm{bd}(D)$. Then we have $\mathfrak{G}[g_\alpha, D] \subseteq G$ and the convex envelope of g_α on D consists of minimizing segments w.r.t G.*

Proof. Analogous to Corollary 4.6. □

We divide G into four sets given by

$$G_1 := \{l_1\} \times [l_2, u_2], \qquad G_2 := [l_1, u_1] \times \{l_2\},$$
$$G_3 := [l_1, u_1] \times \{u_2\}, \qquad G_4 := \{u_1\} \times [l_2, u_2].$$

Again, every minimizing segment $\mathcal{S}_{g_\alpha,G}(\bar{x})$ is defined by its two extreme points p^1 and p^2 with $p^1 \in G_i$ and $p^2 \in G_j$ for some $i, j \in \{1, 2, 3, 4\}$. We classify the minimizing segments for all possible combinations of i and j (with exploited symmetry). Similar to Section 4.2.1, we can exclude the following "easy" cases.

$i = j$: As g_α is convex on G_k for $k = 1, \dots, 4$, we obtain $p^1 = p^2 = \bar{x}$ in this case.

$(i, j) = (1, 3)$: Leads to $p^1 = p^2 = (l_1, u_2)$ by considering the concave directions and Observation 3.8.

$(i, j) = (2, 4)$: Leads to $p^1 = p^2 = (u_1, l_2)$ by considering the concave directions and Observation 3.8.

$(i, j) = (1, 4)$: See Case (1.b).

$(i, j) = (2, 3)$: See Case (2.a).

We consider the combination $(i, j) = (1, 2)$ in more detail. Every minimizing segment is defined by two points $p^1 := (l_1, t)$ and $p^2 := (q, l_2)$. Furthermore, the following equation

$$(l_1 - q)\frac{\partial g_\alpha}{\partial x_1}(q, l_2) + g_\alpha(q, l_2) = (l_2 - t)\frac{\partial g_\alpha}{x_2}(l_1, t) + g_\alpha(l_1, t)$$

must hold. For $q < 0$ we derive

$$\sqrt{\alpha_2 - \alpha_1}\,(q - l_1) = \sqrt{\alpha_2 + \alpha_3}\,(t - l_2),$$

and for $q \geq 0$ we derive

$$(\alpha_1 + \alpha_2)(q - l_1)^2 - 2\alpha_1 l_1^2 = (\alpha_2 + \alpha_3)(t - l_2)^2.$$

Minimizing segments with $p^1 \in G_3$ with $p^2 \in G_4$ are handled analogously. They are not shown in the figures below in order to keep the presentation clean. Again, we derive a possible minimizing segments for several combinations of $i, j \in \{1, \dots, 4\}$ and choose the one with the lowest induced value (see Section 4.2.1).

This results in two possible options for the structure of the convex envelope. The first one consists of, roughly speaking, minimizing segments running from the left to the right side of the box D (see Figure 4.8). Red and blue lines indicate minimizing segments with extreme points in G_1 and G_4, similar to Case (1.b). Green and magenta lines indicate minimizing segments with extreme points in G_1 and G_2, as distinguished above.

The second option instead consists of, roughly speaking, minimizing segments running from the top to the bottom side of the box (see Figure 4.9). Magenta, red and blue lines indicate minimizing segments with extreme points in G_2 and G_3, as described in Case (2.a). Green lines again indicate minimizing segments with extreme points in G_1 and G_2.

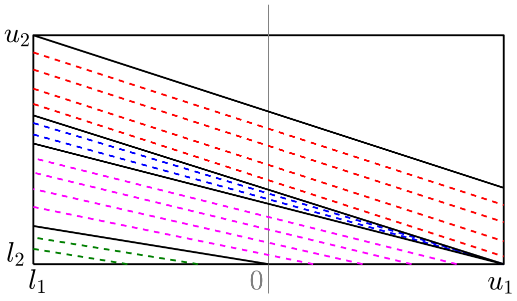


Figure 4.8: Case (2.b.i): Option 1 for the structure of the segments.

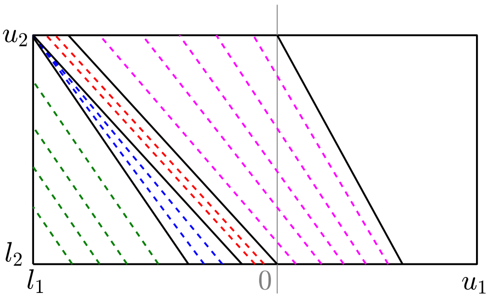


Figure 4.9: Case (2.b.i): Option 2 for the structure of the segments.

Case (2.b.ii)

Function g_α is convex on D. The convex envelope is given by g_α itself.

Case (3.a)

As explained in Section 4.2.2, we first derive the convex envelope w.r.t. $\gamma_{D,1}[g_\alpha]$. In this case, $\gamma_{D,1}[g_\alpha]$ is direction-wise convex w.r.t. component 1 and direction-wise concave w.r.t. component 2. We apply a similar approach as in Case (2.a).

We again derive the structure of minimizing segments by analyzing the derivatives of $\gamma_{D,1}[g_\alpha]$.

As a second step, Figure 4.10 displays the structure of the minimizing segments w.r.t. $\text{vex}_D[\gamma_{D,i}[g_\alpha]]$. Green, yellow and blue lines indicate minimizing segments that are determined by the extreme points $(l_1, u_2), (q, l_2)$ and (u_1, l_2) respectively. The red lines indicate minimizing segments parallel to the vector v.

It is

$$\begin{aligned}
v &:= \left(\tfrac{\alpha_2}{\alpha_1+\alpha_2}, -1\right),\\
s &:= \min\left(u_x,\, l_x\left(1-\sqrt{1-\tfrac{\alpha_2-\alpha_1}{\alpha_2+\alpha_1}}\right)\right),\\
q &:= \min\left(u_x,\, l_x\left(1-\sqrt{1-\tfrac{\alpha_2-\alpha_1}{\alpha_2+\alpha_1}}\right)+\tfrac{\alpha_2}{\alpha_1+\alpha_2}(u_y-l_y)\right).
\end{aligned}$$

Note that the analysis and visualization only holds for $\alpha_2 \geq 0$. However, the case of $\alpha_2 < 0$ is mostly symmetric and can be handled analogously.

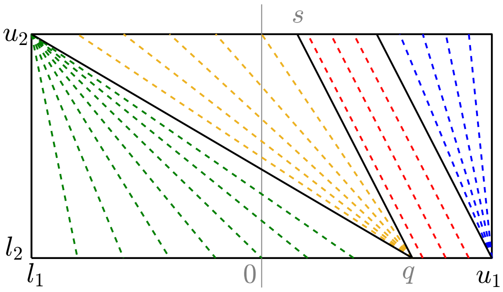


Figure 4.10: Case (3.a): Structure of the minimizing segments.

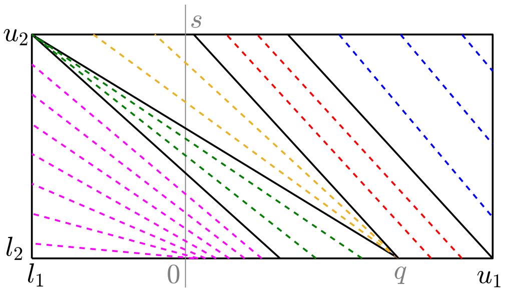


Figure 4.11: Case (3.b.i): Structure of the minimizing segments.

Case (3.b.i)

As explained in Section 4.2.2, we first derive the convex envelope w.r.t. $\gamma_{D,1}[g_\alpha]$. The function $\gamma_{D,1}[g_\alpha]$ is indefinite and direction-wise convex w.r.t components 1 and 2. Hence, we can use the same arguments as in Case (2.b.i) to derive the structure of the minimizing segments.

As a second step, Figure 4.11 displays the structure of the minimizing segments w.r.t. $\text{vex}_D[\gamma_{D,i}[g_\alpha]]$. Minimizing segments connecting opposite sides are colored in green, yellow and red (similar to Case (3.a)). Magenta and blue lines indicate segments that connect adjacent sides. Note that the analysis and visualization only holds for $\alpha_2 \geq 0$. However, the case of $\alpha_2 < 0$ is mostly symmetric and can be handled analogously.

Case (3.b.ii)

As explained in Section 4.2.2, we first derive the convex envelope w.r.t. $\gamma_{D,1}[g_\alpha]$. The function $\gamma_{D,1}[g_\alpha]$ is direction-wise convex w.r.t. components 1 and 2. Furthermore, it is indefinite for $x < s$ as motivated in Section 4.2.2. We use the same strategy and the same subdivision of G as in Case (2.b.iii).

As a second step, Figure 4.12 displays the structure of the minimizing segments w.r.t. $\text{vex}_D[\gamma_{D,i}[g_\alpha]](x)$. Magenta and red lines show segments connecting G_1 and G_4 and yellow lines show segments defined by the extreme point (l_1, u_2). Black dots indicate that the function g_α coincides with its convex envelope. Note that the analysis and visualization only holds for $\alpha_2 \geq 0$. However, the case of $\alpha_2 < 0$ is mostly symmetric and can be handled analogously.

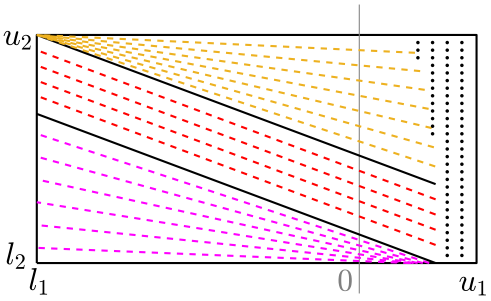


Figure 4.12: Case (3.b.ii): Structure of the minimizing segments.

Concluding the analysis, we are able to determine a minimizing segment of g_α on D for any point $\bar{x} \in D$ and for arbitrary $\alpha \in \mathbb{R}^3$ with very little computational effort. The point $\bar{x}$ is then given as a convex combination of the extreme points of its minimizing segment. The value of $\text{vex}_D[g_\alpha]$ at $\bar{x}$ can be computed by the equivalent convex combination of the values of g_α at these extreme points (see (3.1)). This directly allows us to compute the objective value and a subgradient of the corresponding Separation Problem 3.15 for a given $\alpha \in B^3$ and for all $(\bar{x}, \bar{z}) \in D \times \mathbb{R}^2$ (see Corollary 3.19). Separation Problem 3.15 can now be solved by optimization methods for non-differentiable problems that rely on value and subgradient at every iteration step (see Section 3.2.3). Combining all of this, we can solve the associated Separation Task 3.23 for gas network constraints on a single junction efficiently. Computational results for some test networks are given in the next section.

4.3 Computational Results

In this section, we evaluate the proposed separation strategy from Section 3.2 exemplarily on the feasible set arising from two test networks. We aim to show that the resulting cutting planes are well suited to tighten the convex relaxation of the feasible set provided by state-of-the-art software packages. Additionally, we show that the computation of cutting planes is not very time consuming in comparison to their benefits. We present the test setting in Section 4.3.1, the strategy of our implementation in Section 4.3.2, and discuss the computational results in Section 4.3.3. Section 4.3.4 deals with the separation method based on the Approximate Problem 3.30, that uses an estimator of the convex envelope (see Section 3.3).

This is joint work. The test setting, visualizations, and parts of the computations are not provided by the author. We also refer to [Merkert, 2017].

4.3.1 Test Setting

We consider two test networks. The first one is artificially designed and denoted by "Net1" (see Figure 4.13). It has 7 nodes, 9 arcs and 3 interior junctions. The topology of the second one is taken from a gas network library (GasLib-11, Schmidt et al. [2017], see Figure 4.14) and denoted by "Net2". It has 11 nodes, 11 arcs and 5 interior junctions.

For both networks we consider three different settings each, given by different bounds (box constraints) on the involved variables. We further consider ten different objective functions respectively, that are all minimized in our computations. These objectives are given as linear combinations of the pressure and flow variables in the network. They are either inspired by applications or designed randomly in order to evaluate the relaxed feasible set in multiple "directions". There are 30 different combinations of bound setting and objective function for both networks. We call each combination a scenario.

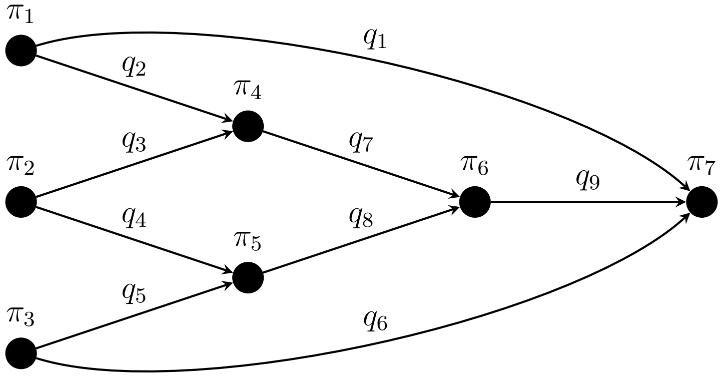


Figure 4.13: Visualization of Net1

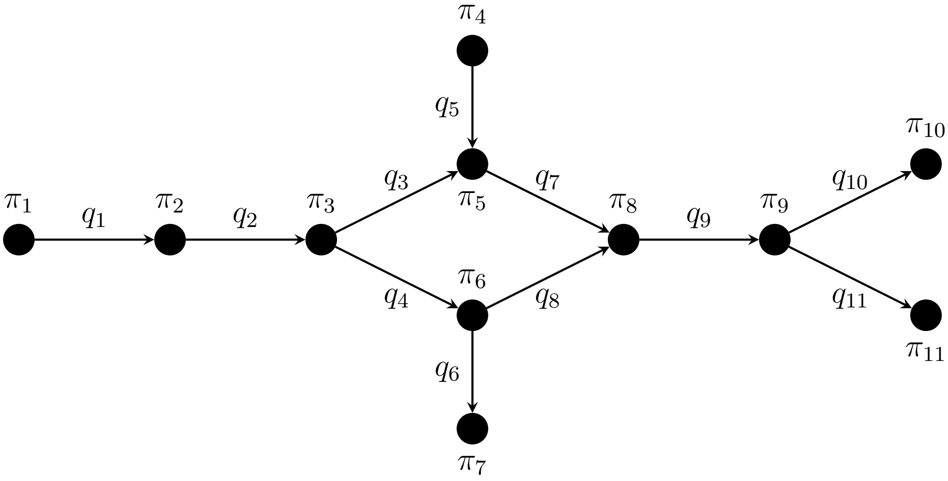


Figure 4.14: Visualization of Net2

The formulation of every interior junction in the considered networks is adapted according to Section 4.1, and has therefore the desired structure as the graph of a vector-valued function (see Problem 3.11). The bounds on the variables are chosen in such a way that several flow directions are fixed automatically. At most one flow direction at every junction remains unfixed. This allows us to compute the convex envelope of all linear combinations of the constraint functions for every junction (see Section 4.2).

As a result, we are able to perform Separation Task 3.23 on the feasible set of every single junction. Note that not the whole network has the desired structure of Problem 3.11, as additional constraints are needed to describe the coupling between the junctions. Therefore we are not able to separate from the convex hull of the feasible set of the whole network, but only from the convex hull of subsets. The design of the implementation is described in the next section.

4.3.2 Implementation

The strategy of our computations is the following. We design and solve a linear relaxation of the network. For each interior junction in the network, we perform Separation Task 3.23 by solving Separation Problem 3.15. This way, we either confirm the solution of the relaxation or derive a cutting plane that is added to the description of the relaxation. We iterate this procedure until no further cutting planes are found, or until a fixed number of iterations is reached.

The Separation Problem 3.15 is implemented as a simple subgradient method. We use an arbitrary $\alpha \in \text{bd}(B^3)$ as a starting point and compute value and subgradient according to Section 4.2 and Corollary 3.19. We make use of a diminishing step size and a stopping criteria based on iteration count and improvement of the objective function. We also apply several standard methods to avoid numerical issues. Note that this part of the implementation is not optimized in terms of computational efficiency, as the focus lies on the quality of the resulting cutting planes. For instance, it could be beneficial to consider a different starting point, like the optimal point of a prior iteration. Furthermore, other optimization methods for non-differentiable problems like bundle methods could be applied (see Section 3.2.3).

If we only consider the progress of the objective value in the iterative linear relaxation outlined above, we simply confirm that the cutting planes hold additional information compared to the linear relaxation. However, our aim is to show that the cutting planes also tighten the "standard" relaxation provided by a state-of-the-art solver for MINLPs. We chose BARON 18.5.8 (Tawarmalani and Sahinidis [2005]) for this comparison. BARON does not allow the user to interfere with the solution process or to integrate custom optimization techniques. Therefore, we add the computed cutting planes to the model description and let BARON solve the problem with and without these additional constraints. We deactivate presolving routines and primal heuristics, and directly provide an optimal solution to the solver. This way, we are able to analyze the influence of our separation strategy on the quality of the convex relaxation and the resulting lower bounds.

We further deactivate the bound tightening strategies provided by BARON, as they are also not available for the iterative linear relaxations used to derive the cutting planes. Otherwise, the cutting planes would be applied on a different relaxed feasible set than the one they are constructed for.

Except for the points above, we choose the default options for BARON. All computations are carried out on a 2.6 GHz Intel Xeon E5-2670 Processor with a limit of 32 GB memory space for each run.

4.3.3 Results

We first present the results of the iterative linear relaxation for both networks. Note that the iterative linear relaxation is in our setting only used to derive the cutting planes. We are not interested in comparing the quality of the linear relaxation with BARON, but in analyzing the influence of the generated cuts on the quality of the lower bounds obtained by BARON as a stand-alone solver. Therefore, we omit any further information on the solution process of the linear relaxation. For all considered scenarios, Table 4.2 and 4.3 display the number of created cuts and the computation time needed for the construction of all cuts combined.

In a second step, we present the results obtained by BARON (see Table 4.4 and 4.5). In column 2 and 3, we display the optimal value of the respective scenario and the lower bound at the root node obtained by BARON alone. Their difference is denoted as the *gap*. For all further settings, we display the respective lower bounds in terms of the percentage of this gap that was closed by the solver. Column 4 gives the lower bound at the root node obtained by BARON with the use of our cutting planes (w/ cuts). Column 5 and 6 show the lower bound after 10 minutes into the solving process. See column 5 for the results of BARON alone and column 6 for BARON with the additional use of our cutting planes.

Scenario	# generated cuts	computation time [s]
1	28	0.22
2	52	0.19
3	50	0.23
4	32	0.12
5	14	0.07
6	34	0.12
7	19	0.09
8	41	0.16
9	38	0.17
10	16	0.07
11	38	0.15
12	25	0.13
13	24	0.08
14	20	0.09
15	25	0.11
16	38	0.09

Table 4.2: Number of generated cutting planes and computation time needed for Net1.

Scenario	# generated cuts	computation time [s]
1	29	0.14
2	24	0.07
3	13	0.05
4	29	0.12
5	13	0.05
6	19	0.03
7	18	0.07
8	32	0.18
9	19	0.07
10	36	0.15
11	18	0.06
12	22	0.06
13	35	0.07
14	52	0.22
15	30	0.09
16	35	0.14
17	37	0.14
18	23	0.1
19	31	0.1

Table 4.3: Number of generated cutting planes and computation time needed for Net2.

Note that several of our 30 scenarios are already solved in the root node. The solution process of these scenarios does not offer any information in terms of improvement of lower bounds. They are therefore excluded from the presentation.

We discuss the artificially designed network Net1 first. 14 of our 30 scenarios are already solved in the root node and are not considered in the tables. The results of the cutting plane generation are given in Table 4.2. Our recursive linear relaxation generates on average 31 cutting planes for every scenario (see columns 2 and 5). We see that the computational effort needed for the construction of the cuts can be neglected. For every scenario, the computation of all cuts together is done in less than a quarter of a second (see columns 3 and 6). The results obtained by BARON are given in Table 4.4. Our generated cuts clearly improve the quality of the lower bounds at the root node for almost all instances (see column 4). This improvement has a large range between 0 % and 84 %, and a mean value of 37 %. After 10 minutes into the solution process, the lower bounds obtained with the cuts are still significantly better than the ones obtained without them (see columns 5 and 6). The average values are 13 % and 68 % respectively, which is a difference of 55 percentage points. Note that 100 % gap closed means that the respective scenario 3 is solved to optimality thanks to the cutting planes.

Next, we discuss the library network Net2. 11 scenarios are already solved in the root node. The results of cutting plane generation are given in Table 4.3. The average number of generated cuts is 27 in this case (see columns 2 and 5) and the construction of cuts is again performed in under a quarter of a second (see columns 3 and 6). The results obtained by BARON are given in Table 4.5. Our observations for Net2 are similar to the ones for Net1. The improvement of the lower bounds at the root node has a large range and a mean value of 54 % (see column 4). After 10 minutes into the solution process, the average amount of gap closed is 17 % without the cuts and 69 % with cuts (see columns 5 and 6). Three scenarios (13, 16 and 19) are solved to optimality within 10 minutes thanks to the cutting planes.

Scenario	Optimal	Root Node		After 10 min.	
	Value	Lower Bound	Gap closed	Gap closed	Gap closed
		by BARON	w/ cuts	by BARON	w/ cuts
1	1051	46	59 %	7 %	66 %
2	-2084	-2500	6 %	0 %	64 %
3	1205	1000	0 %	1 %	100 %
4	-512	-2206	53 %	26 %	63 %
5	1965	-1394	0 %	7 %	8 %
6	402	-2042	84 %	37 %	95 %
7	924	4	54 %	0 %	64 %
8	-2043	-2500	27 %	0 %	81 %
9	-512	-2167	46 %	32 %	82 %
10	2281	-1190	0 %	18 %	18 %
11	441	-2042	84 %	14 %	99 %
12	725	325	53 %	5 %	60 %
13	-1406	-1600	0 %	0 %	57 %
14	-389	-1748	48 %	22 %	68 %
15	2664	-132	0 %	14 %	66 %
16	290	-995	84 %	18 %	98 %

Table 4.4: Improvement of the lower bound for Net1, comparing BARON alone and BARON with the use of the cutting planes

Scenario	Optimal	Root Node		After 10 min.	
	Value	Lower Bound	Gap closed	Gap closed	Gap closed
		by BARON	w/ cuts	by BARON	w/ cuts
1	2139	1812	82 %	6 %	84 %
2	-3837	-4398	52 %	14 %	84 %
3	-53	-360	25 %	0 %	92 %
4	-53	-200	15 %	0 %	31 %
5	460	0	25 %	0 %	83 %
6	531	-2000	98 %	70 %	99 %
7	1639	1635	0 %	30 %	30 %
8	-3853	-4430	50 %	0 %	70 %
9	-10	-210	0 %	0 %	0 %
10	-66	-160	0 %	0 %	4 %
11	-53	-150	0 %	0 %	0 %
12	19	-2170	92 %	3 %	96 %
13	-180	-300	100 %	60 %	100 %
14	2655	2072	85 %	53 %	97 %
15	-3573	-4414	73 %	9 %	89 %
16	21	-248	97 %	0 %	100 %
17	-76	-172	59 %	0 %	66 %
18	399	-31	73 %	4 %	87 %
19	938	-1490	95 %	79 %	100 %

Table 4.5: Improvement of the lower bound for Net2, comparing BARON alone and BARON with the use of the cutting planes

We conclude that our separation method for this special application can be performed in a fraction of a second. This result is expected, as the value and a subgradient of our Separation Problem 3.15 can be derived "mostly" analytically (see Section 4.2). Additionally, the designed cuts are well suited to improve the convex relaxation of the considered MINLP and the resulting lower bound. Furthermore, the amount of gap closed after 10 minutes is higher with the usage of the cuts for every single scenario. This indicates that the growth of the model formulation caused by the additional constraints is not significant compared to the provided benefits of the cuts. We assume that the separation strategy is even more efficient if it is integrated into a MINLP solver, as the cutting planes can be designed adaptively in this setting.

4.3.4 Results by Estimation

In the following, we present computational results that are derived by substituting the convex envelope in Separation Problem 3.15 by a convex underestimator. We make use of the adjusted discretization as presented in Section 3.3 and apply the separation strategy relying on the Approximate Problem 3.30. Similar to the section above, we compute cutting planes using an iterative linear relaxation and add them to the model formulation in BARON. In order to compare the results, we denote the setting that uses the exact Separation Problem 3.15 by *EXACT* (see Sections 4.3.1 – 4.3.3), and the setting in this section that uses the Approximate Problem 3.30 by *APPROX*.

Recall that the computation of the adjusted discretization relies on the solution of a linear optimization problem, while the convex envelope can be determined "mostly" analytically in our case. Furthermore, the quality of the resulting cutting planes is in general better for the exact Separation Problem than for the Approximate Problem. It is therefore expected that *APPROX* needs more time to compute cutting planes and that the improvement of the lower bounds is worse compared to *EXACT*. The benefit of the approximate version on the other hand is its wide applicability. However, here we apply both approaches to the same test instances and use the results of *EXACT* as a reference point to evaluate the computational behavior of *APPROX*.

As a test set for *APPROX*, we consider the artificial network Net1 and its scenarios as described in the previous sections. In order to derive the adjusted discretization for the considered functions, we follow the approach given in Section 3.3 and Example 3.40. A crucial decision in this approach is the accuracy of the discretization and the resulting subdivision of the feasible set. A smaller subdivision leads to more computational effort for the construction of cuts, as the linear subproblems are larger. On the other hand, this leads to tighter convex underestimators and therefore to "stronger" cutting planes and to a greater improvement of the lower bounds. The computation time needed for the generation of cutting planes and the resulting improvement of lower bounds are therefore directly related.

In order to analyze this relation, we consider three different discretizations with increasing accuracy. The corresponding settings are called *APPROX1*, *APPROX2* and *APPROX3*. For *APPROX1*, the feasible set is only divided into approximately 10 sub-boxes. This setting is designed to allow for a generation of cutting planes as fast as given by *EXACT*. For *APPROX3*, the feasible set is divided into approximately 10000 sub-boxes. This setting is designed to generate tight estimations of the convex envelope, and to compute cutting planes with a similar quality as given by *EXACT*. Finally, *APPROX2* is designed as an intermediate setting between the two above. The feasible set is divided into approximately 100 sub-boxes. We expect a trade-off between computation time and quality of cutting planes. Note that the exact number of sub-boxes in these settings may vary due to different interval sizes and the required property of the sub-boxes discussed in Section 3.3.

See Table 4.6 for the results of the cutting plane generation. It displays the number of generated cutting planes and the required computation time for all cuts combined for *EXACT*, *APPROX1*, *APPROX2* and *APPROX3*, respectively. For *APPROX1*, we see that the overall computation time is in fact close to the one for *EXACT* (on average 0.25 s compared to 0.13 s, see columns 3 and 5). The amount of cutting planes on the other hand is significantly lower (on average 4 compared to 31, see columns 2 and 4). This can be explained, as the cutting planes are in general worse and therefore there are fewer cuts able to separate given points. However, we derive that the generation of every single cutting plane still requires significantly more computation time, even for the considered small number of sub-boxes. As expected, the number of generated cutting planes and the computation time are increasing for *APPROX2* and

APPROX3. The average numbers are 24 and 28 (see columns 6 and 8), and the average times are 2 s and 84 s (see columns 7 and 9), respectively. Note in particular, that the computation time for *APPROX3* is unreasonable high for the size of our test instances.

The lower bound obtained by BARON alone is compared to the lower bounds obtained by BARON using the cutting planes generated in the different settings. The relative improvement is again evaluated at the root node and after 10 minutes into the solution process. The results for *APPROX* (compared to *EXACT*) are shown in Table 4.7 for the root node and in Table 4.8 for 10 minutes into the solution process. In both cases, we display the percentage of gap closed by additionally using the cutting planes of *EXACT*, *APPROX1*, *APPROX2* and *APPROX3* respectively. For the meaning of "Gap closed" and the values of the optimal solution and the lower bound obtained by BARON alone, we refer to Section 4.3.3 and Table 4.4.

We consider the lower bounds at the root node first (see Table 4.7). We see that *APPROX1* is only able to improve the lower bounds of three instances and leads to an average improvement of 6 % (see column 3). This is expected as a result of the small number of sub-boxes used for the approximation. *APPROX2* already improves the lower bound of 7 instances and leads to an average improvement of 20 % (see column 4). *APPROX3* leads to similar results as *EXACT*. They both improve the lower bounds of the same instances and the average improvement is given by 36 % and 37 % respectively (see columns 2 and 5).

We derive similar observations for the lower bounds after 10 minutes into the solution process (see Table 4.8). *APPROX1* leads to an average improvement of the lower bound of 24 % (see column 4), which is still better than the average improvement of 13 % given by BARON alone (see column 2). *APPROX2* and *APPROX3* lead to an average improvement of 53 % and 67 % respectively. The latter value is again very close to the average improvement of 68 % obtained by *EXACT*. Two details are worth to mention. First of all, the improvement of scenario 9 is worse for *APPROX1* than for BARON alone. Second, the improvement of scenario 3 is worse for *APPROX2* than for *APPROX1* and for BARON alone. We assume that this is due to the larger model formulation and differences in the solution behavior.

	EXACT		*APPROX1*		*APPROX2*		*APPROX3*	
Scn.	# cuts	time [s]	# cuts	time [s]	# cuts	time [s]	# cuts	time [s]
1	28	0.22	3	0.25	15	1.24	23	76.23
2	52	0.19	4	0.18	41	3.48	36	95.81
3	50	0.23	5	0.38	41	2.56	58	199.08
4	32	0.12	2	0.19	21	1.3	27	60.51
5	14	0.07	1	0.13	14	1.85	17	78.39
6	34	0.12	16	0.76	47	3.44	38	125.97
7	19	0.09	3	0.22	22	1.86	17	51.64
8	41	0.16	6	0.36	47	2.79	36	77.33
9	38	0.17	3	0.24	17	1.52	28	48.41
10	16	0.07	6	0.23	14	1.94	18	89.47
11	38	0.15	2	0.13	34	3.88	41	103.04
12	25	0.13	0	0.05	14	1.65	28	52.82
13	24	0.08	3	0.25	19	2.01	20	73.09
14	20	0.09	1	0.14	11	1.39	14	29.3
15	25	0.11	0	0.05	10	1.09	15	41.56
16	38	0.09	7	0.36	24	3.16	39	142.38

Table 4.6: Number of generated cutting planes and computation time needed for *APPROX* on Net1 (compared to *EXACT*).

Scenario	Gap closed at root node by			
	EXACT	*APPROX1*	*APPROX2*	*APPROX3*
1	59 %	0 %	31 %	54 %
2	6 %	0 %	0 %	5 %
3	0 %	0 %	0 %	0 %
4	53 %	0 %	21 %	50 %
5	0 %	0 %	0 %	0 %
6	84 %	44 %	77 %	83 %
7	54 %	0 %	20 %	51 %
8	27 %	0 %	0 %	21 %
9	46 %	0 %	28 %	44 %
10	0 %	0 %	0 %	0 %
11	84 %	34 %	73 %	83 %
12	53 %	0 %	0 %	51 %
13	0 %	0 %	0 %	0 %
14	48 %	0 %	0 %	44 %
15	0 %	0 %	0 %	0 %
16	84 %	17 %	70 %	83 %

Table 4.7: Improvement of the lower bound at the root node for *APPROX* on Net1 (compared to *EXACT*).

Scenario	Gap closed after 10 min. by				
	BARON	*EXACT*	*APPROX1*	*APPROX2*	*APPROX3*
1	7 %	66 %	14 %	53 %	79 %
2	0 %	64 %	0 %	48 %	56 %
3	1 %	100 %	25 %	0 %	65 %
4	26 %	63 %	26 %	54 %	58 %
5	7 %	8 %	7 %	36 %	37 %
6	37 %	95 %	76 %	92 %	95 %
7	0 %	64 %	0 %	44 %	69 %
8	0 %	81 %	8 %	88 %	90 %
9	32 %	82 %	4 %	69 %	82 %
10	18 %	18 %	18 %	18 %	18 %
11	14 %	99 %	67 %	95 %	98 %
12	5 %	60 %	7 %	48 %	65 %
13	0 %	57 %	11 %	49 %	40 %
14	22 %	68 %	22 %	51 %	67 %
15	14 %	66 %	14 %	14 %	63 %
16	18 %	98 %	79 %	91 %	98 %

Table 4.8: Improvement of the lower bound after 10 minutes for *APPROX* on Net1 (compared to *EXACT* and BARON alone).

We conclude that the separation strategy that relies on the Approximate Problem 3.30 in general requires more computational effort and leads to smaller improvements of the lower bounds in our setting. For the adjusted discretization, our study supports the obvious assumption that effort and improvement behave contrarily to each other and can be regulated by the accuracy of the discretization. In both directions, we were able to nearly reach the quality of the exact version of the separation method. However, the reason for us to consider the approximate version of the separation method in the first place was its wide applicability. It could be beneficial to analyze the interaction of the approximate version with further applications. We assume this to hold based on the following arguments. First of all, estimations of the convex envelope can be designed in a very flexible way with respect to effort and quality as pointed out above. Second, specific applications may allow for specific estimations that may be easier to compute and more accurate. At last, we want to recall that our implementation is not optimized with respect to the computation time of the separation method. All arguments combined, we see potential for future work in this context.

Chapter 5

Monotonic Reformulation and Bound Tightening for Distillation Column Models

In this chapter, we consider another application for MINLPs and present problem specific relaxation refinement strategies for the corresponding feasible set. To be more precise, we develop a bound tightening strategy for problems arising from the modeling of distillation columns.

Distillation columns are an important tool in chemical process design and are used to separate a mixture into its component parts. During the process, the column is filled with a chemical mixture and heated up. As a result, components with high volatility concentrate in the vapor phase at the top part of the column, while components with low volatility concentrate in the liquid phase at the bottom part. In this work we focus on ideal multi-component distillation columns. The term *ideal* refers to the simplified assumptions on the thermodynamic properties of the mixture, while *multi-component* indicates that the considered mixtures may consist of more than three components. For a general introduction to the topic of thermal separation processes, we refer to [Mersmann et al., 2011].

As distillation columns often dominate the chemical production cost, we are interested in finding a cost optimal design of distillation columns for specific separation tasks. These problems can be formulated as MINLPs (see Problem 2.1). Integer variables arise from certain design decisions and nonlinearities are used to model the cost function and physical properties. Despite recent progress in deterministic global optimization (see Chapter 2), these problems are often very difficult to solve due to high computational effort caused by the complexity of the nonlinearities and a large number of variables and equations. Deterministic global optimization of distillation processes is therefore a very challenging task.

In the literature, rigorous deterministic global optimization of distillation processes is not well-covered. Instead, restrictive model assumptions are used to get so-called short-cut models that are easier to solve. In [Nallasivam et al., 2016] for example, optimal sequencing of multi-component distillation columns is calculated using short-cut models at minimum reflux, i.e., smallest feasible internal vapor and liquid flows.

In this work, we drop these simplifying assumptions and use a so called *tray-by-tray* distillation column model (e.g., see Ballerstein et al. [2015]; Kunde et al. [2016]). The column is discretized into several trays and the behavior of the mixture on every tray and the interaction between the trays is modeled explicitly. This modeling strategy allows for more flexibility compared to short-cut models but leads to an increased computational effort. The effort can be significantly reduced by applying problem-specific relaxation refinement strategies as demonstrated for ideal two-component distillation columns in [Ballerstein et al., 2015]. Therein, it has been shown that the computational effort could be reduced by orders of magnitude using a specific bound tightening strategy. It is based on monotonicity of *molar fractions*, i.e., fractions of mole numbers of individual components relative to the total mole number of all components, throughout the column. The application was demonstrated for a hybrid process combining distillation and crystallization units for the separation of two isomers that are difficult to separate by distillation alone. However, an extension to multi-component mixtures is non-trivial, because molar fractions associated with certain components do typically not show the required monotonic behavior in this case.

As a follow-up, a reformulation of a tray-by-tray distillation column model is presented here to overcome this problem. The reformulation is obtained by aggregation of single components, resulting in a linear transformation of the variables used for molar fractions. For mixtures with ideal liquid and vapor behavior, we prove that the transformed variables show the desired property of monotonicity. This allows us to extend the bound tightening strategy from the two-component case to the multi-component setting. It is based on standard interval arithmetic (e.g., see Hansen et al. [1991]; Ratschek and Rokne [1995]), but uses insight on the problem specific constraint structure to be more effective.

The remainder of the chapter is structured as follows. In Section 5.1, we present the ideal multi-component distillation column model we are working with. In Section 5.2, we derive an alternative model formulation by introducing suitable *aggregated components*. In Section 5.3, we prove that the transformed variables associated with each newly introduced aggregated component fulfill the desired monotonicity property. This is exploited in Section 5.4 in order to extend the bound tightening strategy from the two-component case (Ballerstein et al. [2015]) to the general ideal multi-component case. In Section 5.5, we solve several numerical test instances to global optimality and experimentally analyze the influence of the developed techniques on the running time of MINLP solvers.

This chapter is the result of joint work with Achim Kienle, Christian Kunde and Dennis Michaels. The first two sections are mainly based on the literature and previous work, while the author's contribution is presented in Sections 5.3 – 5.5. The respective results are already published in [Mertens et al., 2018].

5.1 Distillation Column Model

This section presents the considered model. We focus on a tray-by-tray model of a distillation column in *steady state*, i.e., explicit modeling of the single trays with no dependency on time. We assume that the partial vapor pressure of each component is equal to the vapor pressure of the pure component multiplied by its molar fraction and that the relative volatility of all components is constant (ideal liquid and vapor phase). Final assumptions are *total condenser* and *total reboiler* (see below), single liquid feed flow at boiling temperature and *constant molar overflow*, i.e., constant value of the liquid/vapor flow in the top and the bottom half of the column respectively. Notation and model description are

Parameters:	
u	Upper bound on the number of trays (length)
α_i	Constant relative volatility of component i
σ	Position of the split between less and more volatile components
π	Purity requirement on the components
F	Feed molar flow entering the column, given in $\mathrm{mol\,s^{-1}}$

Variables:	
V	Vapor flow streaming upwards through the column, given in $\mathrm{mol\,s^{-1}}$
D	Distillate molar flow withdrawn at the condenser, given in $\mathrm{mol\,s^{-1}}$
B	Bottom molar flow withdrawn at the reboiler, given in $\mathrm{mol\,s^{-1}}$
ν_r	Ratio of downward flow to upward flow in the rectifying section
ν_s	Ratio of upward flow to downward flow in the stripping section
l	Number of trays in the part of the column specified by superscript
β	Binary coupling variable determining the position of the feed tray
x_i/y_i	Liquid/Vapor molar fraction of component i
X_k/Y_k	Liquid/Vapor molar fraction of the aggregated component k

Indices:	
in	Feed flow
dist	Distillate flow
bot	Bottom flow
feed	Feed tray
feed−1	Tray above the feed tray
feed+1	Tray below the feed tray
col	Whole column
rect	Rectifying section
l_r	Tray number in rectifying section
strip	Stripping section
l_s	Tray number in stripping section

Table 5.1: List of parameters, variables and indices used in our model.

basically taken from [Ballerstein et al., 2015] and [Kunde et al., 2016], and adapted if necessary.

We are given a mixture consisting of n single components labeled by $1, \ldots, n$. The order of the components is defined with respect to the boiling point. Here, component 1 is the component with the lowest boiling point and component n refers to the component with the highest boiling point. The composition of a mixture is given in terms of molar fractions. A molar fraction denotes the mole numbers of an individual component relative to the total mole number of all components. Thus, the sum of molar fractions over all components is equal to one at every position of the column, which is known as the *summation conditions.*

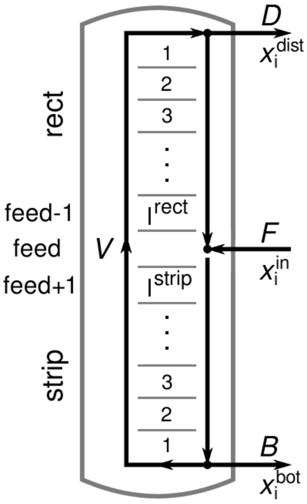


Figure 5.1: A fixed design of a distillation column. Numbers of trays in the rectifying and stripping sections are l^{rect} and l^{strip}. The total number of trays in the column is given by $l^{\text{col}} = l^{\text{rect}} + l^{\text{strip}} + 1$.

Table 5.1 displays the name and meaning of all parameters, variables and indices used in this chapter. A sketch of a distillation column is shown in Figure 5.1. The mixture enters the column at the feed tray with molar feed flow F and initial composition x_i^{in}, $i = 1, \ldots, n$. At the top tray (condenser), the distillate molar flow D leaves the column with composition x_i^{dist}, $i = 1, \ldots, n$, and at the bottom tray (reboiler), the molar flow B leaves the column with composition x_i^{bot}, $i = 1, \ldots, n$. V denotes the vapor flow that streams upwards through the column. The *overall mass balance equations*

$$Fx_i^{\mathrm{in}} = Dx_i^{\mathrm{dist}} + Bx_i^{\mathrm{bot}}, \quad i = 1, \ldots, n \tag{5.1}$$

ensure that the amount of component i entering the column coincides with the overall amount of component i leaving the column.

Rectifying section (above the feed tray) and *stripping section* (below the feed tray) can contain several trays. Trays of the rectifying section are numbered from the top to the bottom by $l_{\mathrm{r}} = 1, \ldots, l^{\mathrm{rect}}$, and trays in the stripping section are numbered from the bottom to the top by $l_{\mathrm{s}} = 1, \ldots, l^{\mathrm{strip}}$. Variables used for molar fractions of component i in liquid and in vapor phases are denoted by x_i and y_i, respectively. We introduce superscripts "feed", "feed$-$1", "feed+1", "rect" and "strip" in order to specify the associated tray of a variable. The script "feed$-$1" denotes the tray above the feed tray and the script "feed+1" denotes the tray below the feed tray. Trays from the rectifying and stripping sections are additionally equipped with their associated tray number as subscript. Mass balances comprising the first tray and a number of consecutive trays are established for the stripping and the rectifying section as well as a mass balance comprising the feed tray. The mass transfer in liquid and vapor phase through the column is then described by the *component mass balance equations*

$$\begin{aligned} y_{i,l_{\mathrm{r}}+1}^{\mathrm{rect}} &= \nu_{\mathrm{r}}\, x_{i,l_{\mathrm{r}}}^{\mathrm{rect}} + (1-\nu_{\mathrm{r}})\, y_{i,1}^{\mathrm{rect}}, \\ \nu_{\mathrm{s}}\, y_i^{\mathrm{feed}} + x_i^{\mathrm{feed}} &= \nu_{\mathrm{s}}\, y_i^{\mathrm{feed}+1} + \nu_{\mathrm{r}}\, \nu_{\mathrm{s}}\, x_i^{\mathrm{feed}-1} + (1-\nu_{\mathrm{r}}\,\nu_{\mathrm{s}})\, x_i^{\mathrm{in}}, \\ x_{i,l_{\mathrm{s}}+1}^{\mathrm{strip}} &= \nu_{\mathrm{s}}\, y_{i,l_{\mathrm{s}}}^{\mathrm{strip}} + (1-\nu_{\mathrm{s}})\, x_{i,1}^{\mathrm{strip}} \end{aligned} \tag{5.2}$$

for $i = 1, \ldots, n$, $l_{\mathrm{r}} = 1, \ldots, u^{\mathrm{rect}}$ and $l_{\mathrm{s}} = 1, \ldots, u^{\mathrm{strip}}$, where u^{rect} and u^{strip} denote upper bounds imposed on l^{rect} and l^{strip}, respectively. We remark that in equations (5.2) the subscripts indicating trays formally range to $u^{\mathrm{rect}} + 1$ and $u^{\mathrm{strip}} + 1$. This way, two artificial trays are introduced to the model. These two trays are later used to model the coupling of the feed tray with the rectifying and the stripping sections (see equations (5.6)). The auxiliary variables $\nu_{\mathrm{r}}, \nu_{\mathrm{s}} \in [0,1]$ defined as

$$\nu_{\mathrm{r}} = \frac{V-D}{V} \quad \text{and} \quad \nu_{\mathrm{s}} = \frac{V}{V+B} \tag{5.3}$$

describe the ratio of upward and downward molar flows in the rectifying and stripping section.

The separation behavior of component i is given by its volatility. Components with higher volatility have a lower boiling point and accumulate in the vapor phase, while components with lower volatility have a higher boiling point and accumulate in the liquid phase. We assume constant relative volatilities of the components, expressed by parameters $\alpha_i > 0$, for $i = 1, \dots, n$. Due to our assumption on the order of the components, we have that $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n$ holds. At all trays, the interactions of the molar fractions in the vapor phase and in the liquid phase are given by the *phase equilibrium equations*

$$y_{i,l_\mathrm{r}}^{\mathrm{rect}} = \frac{\alpha_i \, x_{i,l_\mathrm{r}}^{\mathrm{rect}}}{\sum_{j=1}^n \alpha_j x_{j,l_\mathrm{r}}^{\mathrm{rect}}}, \quad y_i^{\mathrm{feed}} = \frac{\alpha_i \, x_i^{\mathrm{feed}}}{\sum_{j=1}^n \alpha_j x_j^{\mathrm{feed}}}, \quad y_{i,l_\mathrm{s}}^{\mathrm{strip}} = \frac{\alpha_i \, x_{i,l_\mathrm{s}}^{\mathrm{strip}}}{\sum_{j=1}^n \alpha_j x_{j,l_\mathrm{s}}^{\mathrm{strip}}} \tag{5.4}$$

for $i = 1, \dots, n$, $l_\mathrm{r} = 1, \dots, u^{\mathrm{rect}} + 1$ and $l_\mathrm{s} = 1, \dots, u^{\mathrm{strip}} + 1$.

The assumption of a total condenser and a total reboiler are modeled by

$$x_i^{\mathrm{dist}} = y_{i,1}^{\mathrm{rect}} \quad \text{and} \quad x_i^{\mathrm{bot}} = x_{i,1}^{\mathrm{strip}}, \qquad i = 1, \dots, n. \tag{5.5}$$

The total number l^{col} of trays used in a distillation column is given by the number of trays used in the rectifying section, the number of trays used in the stripping section, and the feed tray. To specify l^{col} in our model, the following *coupling conditions* are imposed.

$$\begin{aligned} x_i^{\mathrm{feed}-1} &= \sum_{l_\mathrm{r}=1}^{u^{\mathrm{rect}}} \beta_{l_\mathrm{r}}^{\mathrm{rect}} x_{i,l_\mathrm{r}}^{\mathrm{rect}}, \quad & x_i^{\mathrm{feed}} &= \sum_{l_\mathrm{r}=1}^{u^{\mathrm{rect}}} \beta_{l_\mathrm{r}}^{\mathrm{rect}} x_{i,l_\mathrm{r}+1}^{\mathrm{rect}}, \quad & i &= 1, \dots, n, \\ y_i^{\mathrm{feed}+1} &= \sum_{l_\mathrm{s}=1}^{u^{\mathrm{strip}}} \beta_{l_\mathrm{s}}^{\mathrm{strip}} y_{i,l_\mathrm{s}}^{\mathrm{strip}}, \quad & x_i^{\mathrm{feed}} &= \sum_{l_\mathrm{s}=1}^{u^{\mathrm{strip}}} \beta_{l_\mathrm{s}}^{\mathrm{strip}} x_{i,l_\mathrm{s}+1}^{\mathrm{strip}}, \quad & i &= 1, \dots, n, \end{aligned} \tag{5.6a}$$

$$\begin{aligned} l^{\mathrm{col}} &= \sum_{l_\mathrm{r}=1}^{u^{\mathrm{rect}}} \beta_{l_\mathrm{r}}^{\mathrm{rect}} l_\mathrm{r} + \sum_{l_\mathrm{s}=1}^{u^{\mathrm{strip}}} \beta_{l_\mathrm{s}}^{\mathrm{strip}} l_\mathrm{s} + 1, \\ \sum_{l_\mathrm{r}=1}^{u^{\mathrm{rect}}} \beta_{l_\mathrm{r}}^{\mathrm{rect}} &= 1, \quad \beta_{l_\mathrm{r}}^{\mathrm{rect}} \in \{0,1\}, \;\; l_\mathrm{r} = 1, \dots, u^{\mathrm{rect}}, \\ \sum_{l_\mathrm{s}=1}^{u^{\mathrm{strip}}} \beta_{l_\mathrm{s}}^{\mathrm{strip}} &= 1, \quad \beta_{l_\mathrm{s}}^{\mathrm{strip}} \in \{0,1\}, \;\; l_\mathrm{s} = 1, \dots, u^{\mathrm{strip}}. \end{aligned} \tag{5.6b}$$

Note that the binary variables $\beta_{l_\mathrm{r}}^{\mathrm{rect}}$ and $\beta_{l_\mathrm{s}}^{\mathrm{strip}}$ attain value one if and only if tray l_r of the rectifying section and tray l_s of the stripping section are chosen to be the trays above and below the feed tray in the column.

The purpose of the distillation column is to separate the more volatile components from the less volatile components under given purity constraints. Let the predefined *split* parameter $\sigma \in \{1, \dots, n-1\}$ be the index such that the components 1 to σ belong to the more volatile part and components $\sigma + 1$ to n belong to the less volatile part of the mixture. Let $\pi^{\text{dist}}, \pi^{\text{bot}} \in [0,1]$ further denote the purity requirements imposed on the more volatile components at the condenser and on the less volatile components at the reboiler. Then the *purity constraints* are given as

$$\sum_{i=1}^{\sigma} x_i^{\text{dist}} \geq \pi^{\text{dist}} \quad \text{and} \quad \sum_{i=\sigma+1}^{n} x_i^{\text{bot}} \geq \pi^{\text{bot}}. \tag{5.7}$$

The objective function of our column model reflects the total annualized cost of the distillation process that needs to be minimized. Here, we make use of the following cost function that is taken from previous work (Kunde et al. [2016]).

$$\begin{aligned} \text{cost } &= \lambda_1 V \ + \ \lambda_2 l^{\text{col}} \ + \ \lambda_3(\lambda_4 V \ + \ \lambda_5 B) l^{\text{col}} \\ &\quad + \ \lambda_6(\lambda_4 V \ + \ \lambda_5 B)^{\gamma_1} (\lambda_7 l^{\text{col}} \ + \ \lambda_8)^{\gamma_2} \ + \ \lambda_9(\lambda_4 V \ + \ \lambda_5 B)^{\gamma_3} l^{\text{col}}, \end{aligned} \tag{5.8}$$

with coefficients as specified in Table 5.2. This objective function was originally developed for the distillation of dodecanal and 2-methylundecanal. However, its structure is typical for economical cost estimation and therefore suitable for the computational studies in our work.

Coefficient	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
Value	17544	173.6	2009.7	0.2378	0.0221	2364.5
Coefficient	λ_7	λ_8	λ_9	γ_1	γ_2	γ_3
Value	0.2	4	-171.4	0.533	0.5	0.82

Table 5.2: Coefficients for the cost function

The resulting mixed-integer nonlinear optimization problem is given by

Problem 5.1.

$$\begin{aligned} &\min && (5.8) \\ &\text{s.t.} && (5.1) - (5.7). \end{aligned}$$

We also assume reasonable, real valued bounds on all used variables, that usually arise from the considered application. They are omitted here for a clean presentation but are given for our computational study in Section 5.5. The only discrete variables are $\beta_{l_\mathrm{r}}^{\mathrm{rect}}$ for $l_\mathrm{r} = 1, \dots, u^{\mathrm{rect}}$ and $\beta_{l_\mathrm{s}}^{\mathrm{strip}}$ for $l_\mathrm{s} = 1, \dots, u^{\mathrm{strip}}$, which are binary. They implicitly define the (discrete) length of the distillation column.

5.2 Model Reformulation by Aggregating Components

In the following, we present a reformulation of the given model from Section 5.1. This reformulation is needed to transfer a desired property from the two-component to the multi-component case. For a two-component mixture, molar fractions associated with a single component behave monotonically through the distillation column. Based on this property, a problem-specific bound tightening strategy for two-component distillation column design problems has been developed in previous work (Ballerstein et al. [2015]).

To formalize the notion of monotonicity, we consider the distillation column tray by tray from the bottom to the top. We say that a component i shows *monotonic behavior* through the distillation column when the sequence of respective molar fractions in the liquid phase is either non-decreasing or non-increasing, i. e., either

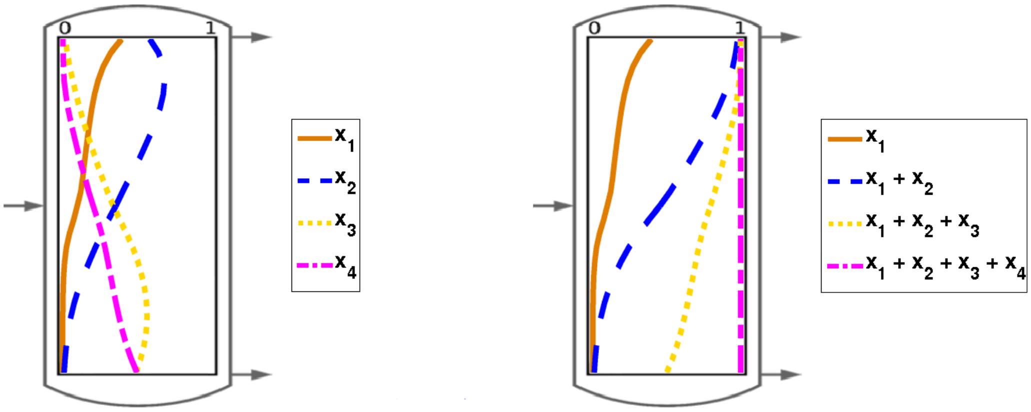
$$x_{i,1}^{\mathrm{strip}} \leq \cdots \leq x_{i,l^{\mathrm{strip}}}^{\mathrm{strip}} \leq x_i^{\mathrm{feed}} \leq x_{i,l^{\mathrm{rect}}}^{\mathrm{rect}} \leq \cdots \leq x_{i,1}^{\mathrm{rect}}$$

or

$$x_{i,1}^{\mathrm{strip}} \geq \cdots \geq x_{i,l^{\mathrm{strip}}}^{\mathrm{strip}} \geq x_i^{\mathrm{feed}} \geq x_{i,l^{\mathrm{rect}}}^{\mathrm{rect}} \geq \cdots \geq x_{i,1}^{\mathrm{rect}}$$

holds. A sequence of values associated with liquid phase (or vapor phase) molar fractions of a single component, considered from the bottom to the top, is also referred to as a *molar fraction profile.*

However, the molar fractions of a single component do possibly not behave monotonically in the multi-component setting. Such a typical situation is illustrated in Figure 5.2 (a) for component 2 (blue-colored dashed curve) and component 3 (yellow-colored dotted curve). This fact makes it hard to generalize the bound tightening strategy from the two-component case (Ballerstein et al. [2015]) to the multi-component case directly.



(a) Molar fraction profiles of original variables

(b) Molar fraction profiles of aggregated variables

Figure 5.2: Molar fraction profiles of original and aggregated variables in liquid phase for a four component mixture.

We overcome this problem by aggregating, for $k = 1, \ldots, n$, the first k components. This is achieved by summing up the corresponding variables used for molar fractions at every position of the distillation column.

The driving force for separating the components of a mixture using distillation is a difference in the volatilities of the single components. For components sorted by decreasing volatility, any group of the first k components has an effective volatility larger than that of the complementary group of $n - k$ components. Therefore, the same direction of the driving force and thus monotonicity of the aggregated molar fractions is expected over the whole column. Although it is expected, there is no proof of this property in the literature to the best of the author's knowledge.

To be more precise, let x_i and y_i (for $i = 1, \ldots, n$) be the variables used for molar fractions of component i in the liquid and in the vapor phase at an arbitrary position. The associated sub- and superscripts indicating the specific position are omitted for a clean presentation. We label the aggregated components by $k = 1, \ldots, n$ and introduce *aggregated concentration variables* X_k and Y_k for the liquid and vapor phases. The original variables used for molar fractions and the new aggregated concentration variables are linearly linked to each other by the bijective relations

$$X_k = \sum_{i=1}^{k} x_i \quad \text{and} \quad Y_k = \sum_{i=1}^{k} y_i, \quad k = 1, \dots, n. \tag{5.9}$$

We also introduce an (aggregated) component 0 for which the liquid and vapor phase concentrations X_0 and Y_0 are zero at all positions. Hence, we can formulate the inverses to the relations (5.9) by

$$x_i = X_i - X_{i-1}, \quad \text{and} \quad y_i = Y_i - Y_{i-1}, \quad k = 1, \dots, n. \tag{5.10}$$

By definition and due to the summation conditions ($\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i = 1$), we have

$$0 = X_0 \leq X_1 \leq \cdots \leq X_n = 1 \quad \text{and} \quad 0 = Y_0 \leq Y_1 \leq \cdots \leq Y_n = 1. \tag{5.11}$$

Note that the equations (5.3), (5.6b) and the objective function (5.8) do not depend on the molar fractions. Hence, they remain unchanged in our aggregated model formulation. Observe further that the overall mass balance equations (5.1), the component mass balance equations (5.2), the coupling conditions (5.6a), total condenser and total reboiler conditions (5.5) and the purity constraints (5.7) are linear in x and y while using the same coefficients for every $i = 1, \dots, n$ respectively. Therefore, we obtain the corresponding constraints for each aggregated component k by summing up the corresponding conditions associated with the first k single components. Only the phase equilibrium equations (5.4) are not linear in the original concentration variables and need to be adapted by applying the inverse relations (5.10). The aggregated model formulation reads as

- The *aggregated overall mass balance equations*:

$$F X_k^{\text{in}} = D X_k^{\text{dist}} + B X_k^{\text{bot}}, \quad k = 1, \dots, n. \tag{5.12}$$

- The aggregated component mass balance equations:

$$\begin{aligned} Y_{k,l_\mathrm{r}+1}^{\text{rect}} &= \nu_\mathrm{r}\, X_{k,l_\mathrm{r}}^{\text{rect}} + (1-\nu_\mathrm{r})\, Y_{k,1}^{\text{rect}}, \\ \nu_\mathrm{s}\, Y_k^{\text{feed}} + X_k^{\text{feed}} &= \nu_\mathrm{s}\, Y_k^{\text{feed}+1} + \nu_\mathrm{r}\, \nu_\mathrm{s}\, X_k^{\text{feed}-1} + (1-\nu_\mathrm{r}\,\nu_\mathrm{s})\, X_k^{\text{in}}, \\ X_{k,l_\mathrm{s}+1}^{\text{strip}} &= \nu_\mathrm{s}\, Y_{k,l_\mathrm{s}}^{\text{strip}} + (1-\nu_\mathrm{s})\, X_{k,1}^{\text{strip}} \end{aligned} \tag{5.13}$$

for $k = 1, \dots, n$, $l_\mathrm{r} = 1, \dots, u^{\text{rect}}$ and $l_\mathrm{s} = 1, \dots, u^{\text{strip}}$.

- The auxiliary variables constraints:

$$\nu_{\mathrm{r}} = \frac{V-D}{V} \quad \text{and} \quad \nu_{\mathrm{s}} = \frac{V}{V+B}. \tag{5.14}$$

- The aggregated phase equilibrium equations:

$$\begin{aligned}
Y_{k,l_{\mathrm{r}}}^{\mathrm{rect}} &= \frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_{\mathrm{r}}}^{\mathrm{rect}} - X_{j-1,l_{\mathrm{r}}}^{\mathrm{rect}})}{\sum_{j=1}^{n} \alpha_j (X_{j,l_{\mathrm{r}}}^{\mathrm{rect}} - X_{j-1,l_{\mathrm{r}}}^{\mathrm{rect}})},\\
Y_{k}^{\mathrm{feed}} &= \frac{\sum_{j=1}^{k} \alpha_j \, (X_{j}^{\mathrm{feed}} - X_{j-1}^{\mathrm{feed}})}{\sum_{j=1}^{n} \; \alpha_j (X_{j}^{\mathrm{feed}} - X_{j-1}^{\mathrm{feed}})},\\
Y_{k,l_{\mathrm{s}}}^{\mathrm{strip}} &= \frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_{\mathrm{s}}}^{\mathrm{strip}} - X_{j-1,l_{\mathrm{s}}}^{\mathrm{strip}})}{\sum_{j=1}^{n} \alpha_j (X_{j,l_{\mathrm{s}}}^{\mathrm{strip}} - X_{j-1,l_{\mathrm{s}}}^{\mathrm{strip}})}
\end{aligned} \tag{5.15}$$

for $k = 1, \ldots, n$, $l_{\mathrm{r}} = 1, \ldots, u^{\mathrm{rect}} + 1$ and $l_{\mathrm{s}} = 1, \ldots, u^{\mathrm{strip}} + 1$.

- The aggregated constraints for the total condenser/reboiler:

$$X_k^{\mathrm{dist}} = Y_{k,1}^{\mathrm{rect}} \quad \text{and} \quad X_k^{\mathrm{bot}} = X_{k,1}^{\mathrm{strip}}, \qquad k = 1, \ldots, n. \tag{5.16}$$

- The aggregated coupling conditions:

$$\begin{aligned}
X_k^{\mathrm{feed}-1} &= \sum_{l_{\mathrm{r}}=1}^{u^{\mathrm{rect}}} \beta_{l_{\mathrm{r}}}^{\mathrm{rect}} X_{k,l_{\mathrm{r}}}^{\mathrm{rect}}, && k = 1, \ldots, n,\\
X_k^{\mathrm{feed}} &= \sum_{l_{\mathrm{r}}=1}^{u^{\mathrm{rect}}} \beta_{l_{\mathrm{r}}}^{\mathrm{rect}} X_{k,l_{\mathrm{r}}+1}^{\mathrm{rect}}, && k = 1, \ldots, n,\\
Y_k^{\mathrm{feed}+1} &= \sum_{l_{\mathrm{s}}=1}^{u^{\mathrm{strip}}} \beta_{l_{\mathrm{s}}}^{\mathrm{strip}} Y_{k,l_{\mathrm{s}}}^{\mathrm{strip}}, && k = 1, \ldots, n,\\
X_k^{\mathrm{feed}} &= \sum_{l_{\mathrm{s}}=1}^{u^{\mathrm{strip}}} \beta_{l_{\mathrm{s}}}^{\mathrm{strip}} X_{k,l_{\mathrm{s}}+1}^{\mathrm{strip}}, && k = 1, \ldots, n,
\end{aligned} \tag{5.17a}$$

$$\begin{aligned}
l^{\mathrm{col}} &= \sum_{l_{\mathrm{r}}=1}^{u^{\mathrm{rect}}} \beta_{l_{\mathrm{r}}}^{\mathrm{rect}} l_{\mathrm{r}} + \sum_{l_{\mathrm{s}}=1}^{u^{\mathrm{strip}}} \beta_{l_{\mathrm{s}}}^{\mathrm{strip}} l_{\mathrm{s}} + 1,\\
\sum_{l_{\mathrm{r}}=1}^{u^{\mathrm{rect}}} \beta_{l_{\mathrm{r}}}^{\mathrm{rect}} &= 1, \quad \beta_{l_{\mathrm{r}}}^{\mathrm{rect}} \in \{0,1\}, \;\; l_{\mathrm{r}} = 1, \ldots, u^{\mathrm{rect}},\\
\sum_{l_{\mathrm{s}}=1}^{u^{\mathrm{strip}}} \beta_{l_{\mathrm{s}}}^{\mathrm{strip}} &= 1, \quad \beta_{l_{\mathrm{s}}}^{\mathrm{strip}} \in \{0,1\}, \;\; l_{\mathrm{s}} = 1, \ldots, u^{\mathrm{strip}}.
\end{aligned} \tag{5.17b}$$

- The aggregated purity constraints:

$$X_\sigma^{\text{dist}} \geq \pi^{\text{dist}} \quad \text{and} \quad (1 - X_\sigma^{\text{bot}}) \geq \pi^{\text{bot}}. \tag{5.18}$$

- The objective function:

$$\begin{aligned} \text{cost } = & \lambda_1 V \ + \ \lambda_2 l^{\text{col}} \ + \ \lambda_3(\lambda_4 V \ + \ \lambda_5 B) l^{\text{col}} \\ & + \ \lambda_6(\lambda_4 V \ + \ \lambda_5 B)^{\gamma_1} \, (\lambda_7 l^{\text{col}} \ + \ \lambda_8)^{\gamma_2} \\ & + \ \lambda_9(\lambda_4 V \ + \ \lambda_5 B)^{\gamma_3} l^{\text{col}}, \end{aligned} \tag{5.19}$$

  with coefficients as specified in Table 5.2.

The resulting reformulated mixed-integer nonlinear optimization problem is given by

Problem 5.2.

$$\begin{aligned} \min \quad & (5.19) \\ \text{s.t.} \quad & (5.12) - (5.18). \end{aligned}$$

It turns out, that the concentration variables of each aggregated component show the desired monotonic behavior, i.e., the overall molar fraction of all components above each possible split position $\sigma \in \{1, \ldots, n-1\}$ change monotonically throughout the distillation column. This is illustrated in Figure 5.2 (b), and will be proven in the next section.

5.3 Monotonicity of the Aggregated Components

In this section we prove that for each aggregated component, the corresponding concentration variables introduced in Section 5.2 behave monotonically through the distillation column. We refer to a sequence of liquid or vapor phase concentration values of an aggregated component as a *(concentration) profile.*

In what follows, we investigate the restrictions of each such profile to the stripping section and to the rectifying section separately. Section 5.3.1 deals with the stripping section. We show that each profile is non-decreasing when considered from the bottom to the top. For the rectifying section discussed in Section 5.3.2, we first apply a suitable transformation. That transformation traces the profiles restricted to the rectifying section back to the case of profiles restricted to the stripping section. We then conclude that each profile also behaves non-decreasingly in the rectifying section from bottom to the top.

As the coupling conditions (5.17) ensure that, for each profile, the parts restricted to the stripping section and restricted to the rectifying section must coincide at the feed tray, we finally obtain that each profile of an aggregated component behaves monotonically through the whole distillation column.

5.3.1 Monotonicity in the Stripping Section

We omit superscript "strip" and denote by $X := (X_{k,l_\mathrm{s}})_{k=0,\dots,n,\, l_\mathrm{s}=1,\dots,u+1}$ and $Y := (Y_{k,l_\mathrm{s}})_{k=0,\dots,n,\, l_\mathrm{s}=1,\dots,u+1}$ the matrices consisting of all liquid and vapor phase concentration variables w.r.t. aggregated components (including the artificial component zero) and restricted to the stripping section.

With this notation and combining the phase equilibrium equations (5.15) with the component mass balance equations (5.13), we obtain the following subsystem that is satisfied by every feasible solution of our distillation column model from Section 5.2.

$$\begin{aligned} X_{k,l_\mathrm{s}+1} = {} & \nu_\mathrm{s}\frac{\sum_{j=1}^{k}\alpha_j(X_{j,l_\mathrm{s}}-X_{j-1,l_\mathrm{s}})}{\sum_{j=1}^{n}\alpha_j(X_{j,l_\mathrm{s}}-X_{j-1,l_\mathrm{s}})} + (1-\nu_\mathrm{s})X_{k,1}, && k=0,\dots,n, \\ & && l_\mathrm{s}=1,\dots,u, \\ 0 < {} & X_{1,l_\mathrm{s}} \le X_{2,l_\mathrm{s}} \le \dots \le X_{n,l_\mathrm{s}}, && l_\mathrm{s}=1,\dots,u+1, \\ X_{0,l_\mathrm{s}} = {} & 0,\ X_{n,l_\mathrm{s}} = 1, && l_\mathrm{s}=1,\dots,u+1, \\ X \in {} & \mathbb{R}^{(n+1)\times(u+1)}, \quad v_s \in [0,1]. && \end{aligned} \tag{5.20}$$

For our analysis, the following remarks are worth to mention.

- In system (5.20), we impose that all variables X_{k,l_s}, $k \ge 1$, are strictly positive. This assumption can be made without loss of generality. In fact, when $X_{k,l_\mathrm{s}} = 0$ holds for some $k \ge 1$ and some l_s, the recursive formula already implies that the concentration of component k is zero at every position in the stripping section, and, hence, in the entire column. In that case we can exclude component k from our considerations.

- To keep the notation simple, we define the following expressions to denote the denominators appearing in system (5.20).

$$N_{l_\mathrm{s}}(X) := \sum_{j=1}^{n} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}}), \quad l_\mathrm{s} = 1, \dots, u+1.$$

  Note that $N_{l_\mathrm{s}}(X) > 0$ holds for all X such that there is a $\nu_\mathrm{s} \in [0,1]$ with (X, ν_s) being feasible to system (5.20).

- Finally, we observe the identities

$$\textstyle\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}}) = \sum_{j=1}^{k-1} X_{j,l_\mathrm{s}} (\alpha_j - \alpha_{j+1}) + X_{k,l_\mathrm{s}} \alpha_k.$$

  for $k = 1, \dots, n$ and $l_\mathrm{s} = 1, \dots, u+1$, that we will frequently use throughout the proofs.

For a solution (X, ν_s) feasible to system (5.20), we will next show that for each aggregated component $k = 1, \dots, n$, the sequence $\{X_{k,l_\mathrm{s}}\}_{l_\mathrm{s}=1}^{u+1}$ is non-decreasing. More precisely, we will prove a more general statement implying the desired property.

Theorem 5.3. *Let $X \in \mathbb{R}^{(n+1)\times(u+1)}$ and $\nu_s \in [0,1]$ be feasible to system* (5.20) *for some $\alpha \in \mathbb{R}^n$ with $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n > 0$. Then,*

$$\frac{X_{k,l_s+1} - X_{k,l_s}}{X_{k,l_s}} \geq \frac{X_{k+1,l_s+1} - X_{k+1,l_s}}{X_{k+1,l_s}} \tag{5.21}$$

holds for $k = 1, \dots, n-1$ and for $l_s = 1, \dots, u$.

Proof. The statement is proven by induction on l_s. We first consider the case with $l_\mathrm{s} = 1$. For an arbitrary $k \in \{1, \dots, n-1\}$, system (5.20) yields

$$X_{k,2} - X_{k,1} = \nu_\mathrm{s} X_{k,1} \left(\frac{\sum_{j=1}^{k} \alpha_j (X_{j,1} - X_{j-1,1})}{N_1(X)\, X_{k,1}} - 1 \right) \quad \text{and}$$

$$X_{k+1,2} - X_{k+1,1} = \nu_\mathrm{s} X_{k+1,1} \left(\frac{\sum_{j=1}^{k+1} \alpha_j (X_{j,1} - X_{j-1,1})}{N_1(X) X_{k+1,1}} - 1 \right),$$

or equivalently

$$\frac{X_{k,2} - X_{k,1}}{X_{k,1}} = \frac{\nu_\mathrm{s}}{N_1(X)} \left(\frac{\sum_{j=1}^{k} \alpha_j (X_{j,1} - X_{j-1,1})}{X_{k,1}} \right) - \nu_\mathrm{s},$$

$$\frac{X_{k+1,2} - X_{k+1,1}}{X_{k+1,1}} = \frac{\nu_\mathrm{s}}{N_1(X)} \left(\frac{\sum_{j=1}^{k+1} \alpha_j (X_{j,1} - X_{j-1,1})}{X_{k+1,1}} \right) - \nu_\mathrm{s}.$$

Thus, in order to prove our statement for $l_\mathrm{s} = 1$, we show that

$$\frac{\sum_{j=1}^{k} \alpha_j (X_{j,1} - X_{j-1,1})}{X_{k,1}} \geq \frac{\sum_{j=1}^{k+1} \alpha_j (X_{j,1} - X_{j-1,1})}{X_{k+1,1}}.$$

Observing that $X_{k+1,1} \geq X_{k,1}$ and $\sum_{j=1}^{k} \alpha_j (X_{j,1} - X_{j-1,1}) - \alpha_{k+1} X_{k,1} \geq 0$ hold, we obtain

$$\begin{aligned}\frac{\sum_{j=1}^{k+1} \alpha_j (X_{j,1} - X_{j-1,1})}{X_{k+1,1}} &= \frac{\sum_{j=1}^{k} \alpha_j (X_{j,1} - X_{j-1,1}) - \alpha_{k+1} X_{k,1}}{X_{k+1,1}} + \alpha_{k+1} \\ &\leq \frac{\sum_{j=1}^{k} \alpha_j (X_{j,1} - X_{j-1,1}) - \alpha_{k+1} X_{k,1}}{X_{k,1}} + \alpha_{k+1} \\ &= \frac{\sum_{j=1}^{k} \alpha_j (X_{j,1} - X_{j-1,1})}{X_{k,1}},\end{aligned}$$

i.e., for $l_\mathrm{s} = 1$, the statement holds for $k = 1, \ldots, n-1$.

Now assume that, for some $l_\mathrm{s} \geq 1$, the statement is true for each $k = 1, \ldots, n-1$. We will show that for $l_\mathrm{s} + 1$ the statement is then true for each $k = 1, \ldots, n-1$ as well. For this, we define $m_k := \frac{X_{k,l_\mathrm{s}}}{X_{k,l_\mathrm{s}-1}}$ for each $k = 1, \ldots, n$. By our induction hypothesis, we have that $m_k \geq m_{k+1}$ holds for $k = 1, \ldots, n-1$.

Next, the values of the terms $X_{k,l_\mathrm{s}+1} - X_{k,l_\mathrm{s}}$ and $X_{k+1,l_\mathrm{s}+1} - X_{k+1,l_\mathrm{s}}$ are compared. Using the recursive formula and the definition of m_k, we obtain for $X_{k,l_\mathrm{s}+1} - X_{k,l_\mathrm{s}}$ that

$$\begin{aligned}&\nu_\mathrm{s} \left(\frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}{N_{l_\mathrm{s}}(X)} - \frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}-1} - X_{j-1,l_\mathrm{s}-1})}{N_{l_\mathrm{s}-1}(X)} \right) = \\ &\qquad \nu_\mathrm{s} \left(\frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}{N_{l_\mathrm{s}}(X)} - \frac{\sum_{j=1}^{k} \alpha_j \left(\frac{X_{j,l_\mathrm{s}}}{m_j} - \frac{X_{j-1,l_\mathrm{s}}}{m_{j-1}} \right)}{N_{l_\mathrm{s}-1}(X)} \right)\end{aligned} \tag{5.22}$$

holds. Moreover, we have

$$\begin{aligned}\frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}{m_1} &\leq \sum_{j=1}^{k} \alpha_j \left(\frac{X_{j,l_\mathrm{s}}}{m_j} - \frac{X_{j-1,l_\mathrm{s}}}{m_{j-1}} \right) \\ &\leq \frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}{m_k}.\end{aligned}$$

By the intermediate value theorem, there must exist some $\tilde{m} \in [m_k, m_1]$ with

$$\sum_{j=1}^{k} \alpha_j \left(\frac{X_{j,l_\mathrm{s}}}{m_j} - \frac{X_{j-1,l_\mathrm{s}}}{m_{j-1}} \right) = \frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}{\tilde{m}}. \tag{5.23}$$

Combining formula (5.22) with formula (5.23) gives rise to

$$\begin{aligned}
&X_{k,l_\mathrm{s}+1} - X_{k,l_\mathrm{s}} \\
&= \nu_\mathrm{s} \sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}}) \left(\frac{1}{N_{l_\mathrm{s}}(X)} - \frac{1}{\tilde{m}\, N_{l_\mathrm{s}-1}(X)} \right) \\
&= X_{k,l_\mathrm{s}} \nu_\mathrm{s} \frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}{X_{k,l_\mathrm{s}}} \left(\frac{1}{N_{l_\mathrm{s}}(X)} - \frac{1}{\tilde{m} N_{l_\mathrm{s}-1}(X)} \right).
\end{aligned} \tag{5.24}$$

Again, using the recursive formula and the definition of m_k, the second term can be rewritten as

$$\begin{aligned}
&X_{k+1,l_\mathrm{s}+1} - X_{k+1,l_\mathrm{s}} \\
&= \nu_\mathrm{s} \Bigg(\frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}}) + \alpha_{k+1} (X_{k+1,l_\mathrm{s}} - X_{k,l_\mathrm{s}})}{N_{l_\mathrm{s}}(X)} \\
&\qquad - \frac{\sum_{j=1}^{k} \alpha_j \left(\frac{X_{j,l_\mathrm{s}}}{m_j} - \frac{X_{j-1,l_\mathrm{s}}}{m_{j-1}} \right) + \alpha_{k+1} \left(\frac{X_{k+1,l_\mathrm{s}}}{m_{k+1}} - \frac{X_{k,l_\mathrm{s}}}{m_k} \right)}{N_{l_\mathrm{s}-1}(X)} \Bigg) \\
&= \nu_\mathrm{s} \left(\frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}{N_{l_\mathrm{s}}(X)} - \frac{\sum_{j=1}^{k} \alpha_j (\frac{X_{j,l_\mathrm{s}}}{m_j} - \frac{X_{j-1,l_\mathrm{s}}}{m_{j-1}})}{N_{l_\mathrm{s}-1}(X)} \right) \\
&\qquad + \nu_\mathrm{s} \left(\frac{(\alpha_{k+1} (X_{k+1,l_\mathrm{s}} - X_{k,l_\mathrm{s}})}{N_{l_\mathrm{s}}(X)} - \frac{\alpha_{k+1} (\frac{X_{k+1,l_\mathrm{s}}}{m_{k+1}} - \frac{X_{k,l_\mathrm{s}}}{m_k})}{N_{l_\mathrm{s}-1}(X)} \right).
\end{aligned}$$

Using formula (5.23) and the fact that $m_{k+1} \leq m_k \leq \tilde{m}$ holds, we can further estimate

$$\begin{aligned}
&X_{k+1,l_\mathrm{s}+1} - X_{k+1,l_\mathrm{s}} \\
&= \nu_\mathrm{s} \sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}}) \left(\frac{1}{N_{l_\mathrm{s}}(X)} - \frac{1}{\tilde{m} N_{l_\mathrm{s}-1}(X)} \right) \\
&\qquad + \nu_\mathrm{s} \left(\frac{(\alpha_{k+1} (X_{k+1,l_\mathrm{s}} - X_{k,l_\mathrm{s}})}{N_{l_\mathrm{s}}(X)} - \frac{\alpha_{k+1} \left(\frac{X_{k+1,l_\mathrm{s}}}{m_{k+1}} - \frac{X_{k,l_\mathrm{s}}}{m_k} \right)}{N_{l_\mathrm{s}-1}(X)} \right) \\
&\leq \nu_\mathrm{s} \sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}}) \left(\frac{1}{N_{l_\mathrm{s}}(X)} - \frac{1}{\tilde{m} N_{l_\mathrm{s}-1}(X)} \right) \\
&\qquad + \nu_\mathrm{s} \left(\frac{(\alpha_{k+1} (X_{k+1,l_\mathrm{s}} - X_{k,l_\mathrm{s}})}{N_{l_\mathrm{s}}(X)} - \frac{\alpha_{k+1} (X_{k+1,l_\mathrm{s}} - X_{k,l_\mathrm{s}})}{\tilde{m} N_{l_\mathrm{s}-1}(X)} \right) \\
&= \nu_\mathrm{s} \sum_{j=1}^{k+1} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}}) \left(\frac{1}{N_{l_\mathrm{s}}(X)} - \frac{1}{\tilde{m}\, N_{l_\mathrm{s}-1}(X)} \right).
\end{aligned}$$

Finally, we exploit that $X_{k+1,l_\mathrm{s}} \geq X_{k,l_\mathrm{s}}$ and that

$$\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}}) - \alpha_{k+1} X_{k,l_\mathrm{s}} \geq 0$$

holds. This yields

$$\begin{aligned}
&X_{k+1,l_\mathrm{s}+1} - X_{k+1,l_\mathrm{s}} \\
&\leq X_{k+1,l_\mathrm{s}} \nu_\mathrm{s} \left(\frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}}) - \alpha_{k+1} X_{k,l_\mathrm{s}}}{X_{k+1,l_\mathrm{s}}} + \alpha_{k+1} \right) \\
&\qquad \cdot \left(\frac{1}{N_{l_\mathrm{s}}(X)} - \frac{1}{\tilde{m} N_{l_\mathrm{s}-1}(X)} \right) \\
&\leq X_{k+1,l_\mathrm{s}} \nu_\mathrm{s} \left(\frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}}) - \alpha_{k+1} X_{k,l_\mathrm{s}}}{X_{k,l_\mathrm{s}}} + \alpha_{k+1} \right) \\
&\qquad \cdot \left(\frac{1}{N_{l_\mathrm{s}}(X)} - \frac{1}{\tilde{m} N_{l_\mathrm{s}-1}(X)} \right) \\
&= X_{k+1,l_\mathrm{s}} \nu_\mathrm{s} \left(\frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}{X_{k,l_\mathrm{s}}} \right) \left(\frac{1}{N_{l_\mathrm{s}}(X)} - \frac{1}{\tilde{m}\, N_{l_\mathrm{s}-1}(X)} \right).
\end{aligned} \tag{5.25}$$

From formulas (5.24) and (5.25), we can deduce that

$$\frac{X_{k,l_\mathrm{s}+1} - X_{k,l_\mathrm{s}}}{X_{k,l_\mathrm{s}}} \geq \frac{X_{k+1,l_\mathrm{s}+1} - X_{k+1,l_\mathrm{s}}}{X_{k+1,l_\mathrm{s}}}$$

holds. □

We are now able to prove that the concentration profiles behave non-decreasing through the stripping section.

Corollary 5.4. *Let (X, ν_s) be a feasible solution to system* (5.20) *for some $\alpha \in \mathbb{R}^n$ with $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n > 0$. Let Y be the matrix consisting of all vapor phase concentration variables Y_{k,l_s} that are implied by X through the phase equilibrium equations* (5.15). *Then, for each aggregated component $k = 1, \ldots, n$, both sequences $\{X_{k,l_s}\}_{l_s=1}^{u+1}$ and $\{Y_{k,l_s}\}_{l_s=1}^{u+1}$ are non-decreasing.*

Proof. By definition, we have that $X_{n,l_\mathrm{s}} = 1$ holds for $l_\mathrm{s} = 1, \ldots, u+1$. Thus, the statement holds for $k = n$. For each fixed $l_\mathrm{s} \in \{1, \ldots, u\}$, we obtain from Theorem 5.3 that

$$\frac{X_{1,l_\mathrm{s}+1} - X_{1,l_\mathrm{s}}}{X_{1,l_\mathrm{s}}} \geq \frac{X_{2,l_\mathrm{s}+1} - X_{2,l_\mathrm{s}}}{X_{2,l_\mathrm{s}}} \geq \cdots \geq \frac{X_{n,l_\mathrm{s}+1} - X_{n,l_\mathrm{s}}}{X_{n,l_\mathrm{s}}} = 0$$

holds. By assumption $X_{k,l_\mathrm{s}} > 0$, for $k = 1, \dots, n$ and $l_\mathrm{s} = 1, \dots, u+1$, it follows that $X_{k,l_\mathrm{s}+1} - X_{k,l_\mathrm{s}} \geq 0$, for all $k = 1, \dots, n$. This proves the statement for sequence $\{X_{k,l_\mathrm{s}}\}_{l_\mathrm{s}=1}^{u+1}$.

Using the equations (5.13), and $X_{k,l_\mathrm{s}+2} \geq X_{k,l_\mathrm{s}+1}$, for all k and l_s, we can further derive

$$\nu_\mathrm{s} Y_{k,l_\mathrm{s}+1} + (1-\nu_\mathrm{s}) X_{k,1} \geq \nu_\mathrm{s} Y_{k,l_\mathrm{s}} + (1-\nu_\mathrm{s}) X_{k,1} \quad \Leftrightarrow \quad \nu_\mathrm{s}(Y_{k,l_\mathrm{s}+1} - Y_{k,l_\mathrm{s}}) \geq 0$$

for all $k = 1, \dots, n$ and $l_\mathrm{s} = 1, \dots, u^\mathrm{strip}$. This implies that $Y_{k,l_\mathrm{s}+1} - Y_{k,l_\mathrm{s}} \geq 0$ holds for $\nu_\mathrm{s} > 0$. Moreover, if $\nu_\mathrm{s} = 0$, then we can deduce from the equations (5.13) that $X_{k,1} = X_{k,2} = \cdots = X_{k,u+1}$ holds. By phase equilibrium equations (5.15), it follows $Y_{k,1} = Y_{k,2} = \cdots = Y_{k,u+1}$. □

5.3.2 Monotonicity in the Rectifying Section

Next, we prove monotonicity of the profiles restricted to the rectifying section. We omit superscript "rect" and denote the matrices consisting of all liquid and vapor phase concentration variables of aggregated components (including the artificial component zero) and restricted to the rectifying section by $X := (X_{k,l_\mathrm{r}})_{k=0,\dots,n,\, l_\mathrm{r}=1,\dots,u+1}$ and $Y := (Y_{k,l_\mathrm{r}})_{k=0,\dots,n,\, l_\mathrm{r}=1,\dots,u+1}$.

Recall, that in our model description the trays in the rectifying section are labeled from top to bottom. For this labeling, we show that the sequences $\{X_{k,l_\mathrm{r}}\}_{l_\mathrm{r}=1}^{u+1}$ and $\{Y_{k,l_\mathrm{r}}\}_{l_\mathrm{r}=1}^{u+1}$ are non-increasing for every $k = 1, \dots, n$. Therefore, the profiles considered from the bottom to the top are non-decreasing.

In order to derive a system for the rectifying section that corresponds to system (5.20) of the stripping section, we need the well-known inverses of the phase equilibrium equations (5.15). For each aggregated component $k \geq 1$ and for each tray l_r, they are given as

$$X_{k,l_\mathrm{r}} = \frac{\sum_{j=1}^{k} \alpha_j^{-1} (Y_{j,l_\mathrm{r}} - Y_{j-1,l_\mathrm{r}})}{\sum_{j=1}^{n} \alpha_j^{-1} (Y_{j,l_\mathrm{r}} - Y_{j-1,l_\mathrm{r}})}, \quad l_\mathrm{r} = 1, \dots, u+1. \tag{5.26}$$

Using (5.26), we obtain from the component mass balance equations (5.13) the following subsystem

$$\begin{array}{rlll}
Y_{k,l_{\mathrm{r}}+1} = & \nu_{\mathrm{r}}\frac{\sum_{j=1}^{k}\alpha_j^{-1}(Y_{j,l_{\mathrm{r}}}-Y_{j-1,l_{\mathrm{r}}})}{\sum_{j=1}^{n}\alpha_j^{-1}(Y_{j,l_{\mathrm{r}}}-Y_{j-1,l_{\mathrm{r}}})} + (1-\nu_{\mathrm{r}})Y_{k,1}, & k = 0,\dots,n, & \\
 & & l_{\mathrm{r}} = 1,\dots,u, & \\
0 < & Y_{1,l_{\mathrm{r}}} \le Y_{2,l_{\mathrm{r}}} \le \cdots \le Y_{n,l_{\mathrm{r}}}, & l_{\mathrm{r}} = 1,\dots,u+1, & (5.27)\\
Y_{0,l_{\mathrm{r}}} = & 0,\;\; Y_{n,l_{\mathrm{r}}} = 1, & l_{\mathrm{r}} = 1,\dots,u+1, & \\
Y & \in \mathbb{R}^{(n+1)\times(u+1)}, \qquad v_r \in [0,1], & &
\end{array}$$

that must be satisfied by every feasible solution of our distillation column model.

Now, we make use of the following transformation rules.

$$\begin{array}{lll}
\hat{u} := u, \qquad \hat{\nu}_{\mathrm{s}} := \nu_{\mathrm{r}}, & & \\
\hat{Y}_{k,l} := (1 - X_{n-k,l}), & & \\
 & k = 0,\dots,n, \quad l = 1,\dots,u+1, & (5.28)\\
\hat{X}_{k,l} := (1 - Y_{n-k,l}), & & \\
\hat{\alpha}_k := \alpha_{n+1-k}^{-1}, & k = 1,\dots,n. &
\end{array}$$

These rules allow us to restate system (5.27) equivalently as

$$\begin{array}{lll}
\hat{X}_{k,l+1} = \hat{\nu}_{\mathrm{s}}\frac{\sum_{j=1}^{k}\hat{\alpha}_j(\hat{X}_{j,l}-\hat{X}_{j-1,l})}{\sum_{j=1}^{n}\hat{\alpha}_j(\hat{X}_{j,l}-\hat{X}_{j-1,l})} + (1-\hat{\nu}_{\mathrm{s}})\hat{X}_{k,1}, & k = 0,\dots,n, & \\
 & l = 1,\dots,\hat{u}, & \\
0 < \hat{X}_{1,l} \le \hat{X}_{2,l} \le \cdots \le \hat{X}_{n,l}, & l = 1,\dots,\hat{u}+1, & (5.29)\\
\hat{X}_{0,l} = 0,\;\; \hat{X}_{n,l} = 1, & l = 1,\dots,\hat{u}+1, & \\
\hat{X} \in \mathbb{R}^{(n+1)\times(\hat{u}+1)}, \qquad \hat{v}_s \in [0,1]. & &
\end{array}$$

We can draw the following conclusion:

Corollary 5.5. *Let (Y,ν_r) be a feasible solution to system* (5.27) *for some $\alpha \in \mathbb{R}^n$ with $\alpha_1 \ge \alpha_2 \ge \cdots \ge \alpha_n > 0$. Let X be the matrix consisting of all liquid phase concentrations variables X_{k,l_r}, that are implied by Y through the phase equilibrium equations* (5.26)*. Then, for each aggregated component $k \in \{1,\dots,n\}$, both sequences $\{Y_{k,l_r}\}_{l_r=1}^{u+1}$ and $\{X_{k,l_r}\}_{l_r=1}^{u+1}$ are non-increasing.*

Proof. Consider system (5.27) with parameters $\alpha_1 \ge \alpha_2 \ge \cdots \ge \alpha_n > 0$ and variables (Y,ν_{r}). We show that applying the transformation rules (5.28) to (Y,ν_{r}) and $\alpha_1,\dots,\alpha_n$ leads to system (5.29) with parameters $\hat{\alpha}_1 \ge \hat{\alpha}_2 \ge \cdots \ge \hat{\alpha}_n > 0$ and with variables $\hat{X}$, $\hat{\nu}_{\mathrm{s}}$ whose feasible solutions satisfy the conditions of Theorem 5.3 (and Corollary 5.4). Note that feasible solutions to system (5.27) are in one-to-one correspondence to solutions feasible to system (5.29) via the transformation rules (5.28). As we obtain from Corollary 5.4 that, for every feasible solution $(\hat{X}, \hat{\nu}_{\mathrm{s}})$, the sequences $\{\hat{X}_{k,\hat{l}_{\mathrm{s}}}\}_{\hat{l}_{\mathrm{s}}=1}^{\hat{u}+1}$ with $k = 1, \ldots, n$ are non-decreasing, the corresponding sequences $\{Y_{k,l_{\mathrm{r}}}\}_{l_{\mathrm{r}}=1}^{u+1}$, $k = 1, \ldots, n$ are non-increasing.

The first part of system (5.27) is given as a combination of the mass balance equations (5.13) and the inverted phase equilibrium equations (5.26), i.e.,

$$Y_{k,l_{\mathrm{r}}+1} = \nu_{\mathrm{r}}\, X_{k,l_{\mathrm{r}}} + (1-\nu_{\mathrm{r}})\, Y_{k,1} \quad \text{and} \quad X_{k,l_{\mathrm{r}}} = \frac{\sum_{j=1}^{k} \alpha_j^{-1}(Y_{j,l_{\mathrm{r}}} - Y_{j-1,l_{\mathrm{r}}})}{\sum_{j=1}^{n} \alpha_j^{-1}(Y_{j,l_{\mathrm{r}}} - Y_{j-1,l_{\mathrm{r}}})},$$

for $k = 1, \ldots, n$ and $l_{\mathrm{r}} = 1, \ldots, u$. We apply the transformation rules (5.28) to each constraint separately.

For all $k = 1, \ldots, n$ and all $l = 1, \ldots, u$, we obtain

$$\begin{aligned} & Y_{k,l+1} = \nu_{\mathrm{r}} X_{k,l} + (1-\nu_{\mathrm{r}}) Y_{k,1} \\ \Leftrightarrow\ & (1 - Y_{k,l+1}) = 1 - \Big(\nu_{\mathrm{r}} X_{k,l} + (1-\nu_{\mathrm{r}}) Y_{k,1}\Big) \\ \Leftrightarrow\ & (1 - Y_{k,l+1}) = \nu_{\mathrm{r}}(1 - X_{k,l}) + (1-\nu_{\mathrm{r}})(1 - Y_{k,1}). \end{aligned}$$

Thus, the transformation rules (5.28) yield

$$\hat{X}_{k,l+1} = \hat{\nu}_{\mathrm{s}} \hat{Y}_{k,l} + (1-\hat{\nu}_{\mathrm{s}}) \hat{X}_{k,1}, \qquad m = 0, \ldots, n, \quad l = 1, \ldots, u. \tag{5.30}$$

For the inverted phase equilibrium equations, we further derive that

$$\begin{aligned} \hat{Y}_{n-k,l} &= 1 - X_{k,l} = 1 - \frac{\sum_{j=1}^{k} \alpha_j^{-1}(Y_{j,l} - Y_{j-1,l})}{\sum_{j=1}^{n} \alpha_j^{-1}(Y_{j,l} - Y_{j-1,l})} = \frac{\sum_{j=k+1}^{n} \alpha_j^{-1}(Y_{j,l} - Y_{j-1,l})}{\sum_{j=1}^{n} \alpha_j^{-1}(Y_{j,l} - Y_{j-1,l})} \\ &= \frac{\sum_{j=k+1}^{n} \alpha_j^{-1}(1 - \hat{X}_{n-j,l} - \big(1 - \hat{X}_{n-(j-1),l})\big)}{\sum_{j=1}^{n} \alpha_j^{-1}(1 - \hat{X}_{n-j,l} - \big(1 - \hat{X}_{n-(j-1),l})\big)} \\ &= \frac{\sum_{j=k+1}^{n} \alpha_j^{-1}(\hat{X}_{n-(j-1),l} - \hat{X}_{n-j,l})}{\sum_{j=1}^{n} \alpha_j^{-1}(\hat{X}_{n-(j-1),l} - \hat{X}_{n-j,l_{\mathrm{s}}})} = \frac{\sum_{j=k+1}^{n} \hat{\alpha}_{n+1-j}(\hat{X}_{n+1-j,l} - \hat{X}_{n-j,l})}{\sum_{j=1}^{n} \hat{\alpha}_{n+1-j}(\hat{X}_{n+1-j,l} - \hat{X}_{n-j,l})} \end{aligned}$$

holds for every $k = 1, \ldots, n$ and for every $l = 1, \ldots, u$. Recall that the components appear in reverse order after the transformation. To indicate this, we introduce new indices $m := n-k$ and $p := n+1-j$. We derive the equivalence to the non-inverted phase equilibrium equations (5.15) by

$$\begin{aligned} \hat{Y}_{n-k,l} &= \frac{\sum_{p=1}^{n-k} \beta_p(\hat{X}_{p,l} - \hat{X}_{p-1,l})}{\sum_{p=1}^{n} \beta_p(\hat{X}_{p,l} - \hat{X}_{p-1,l})}, \quad && n = 0,\dots,n, \\ & && l = 1,\dots,u+1 \\ \Leftrightarrow \qquad \hat{Y}_{m,l} &= \frac{\sum_{p=1}^{m} \beta_p(\hat{X}_{p,l} - \hat{X}_{p-1,l})}{\sum_{p=1}^{n+1} \beta_p(\hat{X}_{p,l} - \hat{X}_{p-1,l})}, \quad && m = 0,\dots,n, \\ & && l = 1,\dots,u+1. \end{aligned} \tag{5.31}$$

Combining the equations (5.30) and (5.31), we obtain the first line from system (5.29).

The second line of system (5.29) results from the relation

$$Y_{k+1,l} \geq Y_{k,l}, \;\; k = 0,\dots,n-1 \quad \Leftrightarrow \quad \hat{X}_{m-1,l} \leq \hat{X}_{m,l}, \;\; m = 1,\dots,n.$$

The third line of system (5.29) trivially holds.

It remains to argue that the transformed constant relative volatilities $\hat{\alpha}_m = \alpha_{(n+1)-m}^{-1}$ for $m = 1,\dots,n$ are strictly positive and monotonically non-decreasing in the new ordering of the components. In fact, this holds as $\alpha_1 \geq \dots \geq \alpha_n > 0$ implies

$$0 < \alpha_1^{-1} \equiv \hat{\alpha}_n \leq \dots \leq \alpha_n^{-1} \equiv \hat{\alpha}_1.$$

This means that system (5.29) satisfies all conditions of Theorem 5.3 and Corollary 5.4. We conclude that, for every $k = 1,\dots,n$, both sequences $\{\hat{X}_{k,l}\}_{l=1}^{\hat{u}+1}$ and $\{\hat{Y}_{k,l}\}_{l=1}^{\hat{u}+1}$ are non-decreasing. Using transformations rules (5.28) again, the statement follows. □

In summary, Corollary 5.4 and 5.5 guarantee that the variables of every feasible solution to our reformulated model (Problem 5.2) behave monotonically in the rectifying and stripping section. If we take the coupling conditions (5.17) into account, we derive the desired relation for the whole column, i.e.,

$$X_{i,1}^{\text{strip}} \leq \dots \leq X_{i,l^{\text{strip}}}^{\text{strip}} \leq X_i^{\text{feed}} \leq X_{i,l^{\text{rect}}}^{\text{rect}} \leq \dots \leq X_{i,1}^{\text{rect}}$$

and

$$Y_{i,1}^{\text{strip}} \leq \dots \leq Y_{i,l^{\text{strip}}}^{\text{strip}} \leq Y_i^{\text{feed}} \leq Y_{i,l^{\text{rect}}}^{\text{rect}} \leq \dots \leq Y_{i,1}^{\text{rect}}$$

for all $i = 1,\dots,n$.

5.4 Problem Specific Relaxation Refinement

In this section, we use the results from Section 5.3 to derive problem specific relaxation refinement strategies for our distillation column model associated with the aggregated components (Problem 5.2). In particular, we develop a recursive bound tightening strategy in Section 5.4.1. For this, we adapt the arguments used in previous work [Ballerstein et al., 2015] for two-component distillation columns to the multi-component case with aggregated components.

In Section 5.4.2, we moreover restate a method to derive additional bounds on the aggregated concentration variables by computing the fixed points of the concentration profiles. This method has already been applied in [Kunde et al., 2016] to the two-component distillation case. Both techniques are implemented in global optimization software and their impact is computationally evaluated in Section 5.5.

5.4.1 A Recursive Bound Tightening Strategy

The bound tightening strategy that is developed in this paper for ideal multi-component distillation column models is a feasibility based bound tightening (see Section 2.2.1). In particular, we make use of interval arithmetics that we apply to two types of well-structured model constraints.

The monotonic behavior of the aggregated concentration profiles together with the aggregated component mass balance equations allows us to propagate given bounds on the aggregated concentration variables at a certain tray to the aggregated concentration variables associated with an adjacent tray.

For this, recall that for every $k = 1, \ldots, n$, the aggregated component mass balance equations (5.13)

$$\begin{aligned} Y^{\mathrm{rect}}_{k,l_{\mathrm{r}}+1} &= \nu_{\mathrm{r}}\, X^{\mathrm{rect}}_{k,l_{\mathrm{r}}} + (1-\nu_{\mathrm{r}})\, Y^{\mathrm{rect}}_{k,1}, \qquad && l_{\mathrm{r}} = 1, \ldots, u^{\mathrm{rect}}, \\ X^{\mathrm{strip}}_{k,l_{\mathrm{s}}+1} &= \nu_{\mathrm{s}}\, Y^{\mathrm{strip}}_{k,l_{\mathrm{s}}} + (1-\nu_{\mathrm{s}})\, X^{\mathrm{strip}}_{k,1}, \qquad && l_{\mathrm{s}} = 1, \ldots, u^{\mathrm{strip}} \end{aligned}$$

associated with the trays in the rectifying and stripping sections form two families of recursive functions (one for each section). By analyzing the partial derivatives, one can show that in both cases the recursive functions behave monotonically in each of their arguments (see also Ballerstein et al. [2015] for the two-component case). The analysis is mainly straightforward. Only the partial derivatives

$$\frac{\partial Y^{\text{rect}}_{k,l_{\text{r}}+1}}{\partial \nu_{\text{r}}} = X^{\text{rect}}_{k,l_{\text{r}}} - Y^{\text{rect}}_{k,1} \quad \text{and} \quad \frac{\partial X^{\text{strip}}_{k,l_{\text{s}}+1}}{\partial \nu_{\text{s}}} = Y^{\text{strip}}_{k,l_{\text{s}}} - X^{\text{strip}}_{k,1}$$

need special attention. For these, we remark that the monotonicity of the concentration profiles ensures for each aggregated component $k = 1, \ldots, n$ that

$$\begin{aligned} X^{\text{rect}}_{k,l_{\text{r}}} &\leq X^{\text{rect}}_{k,1}, \quad l_{\text{r}} = 1, \ldots, u^{\text{rect}}, \\ Y^{\text{strip}}_{k,l_{\text{s}}} &\geq Y^{\text{strip}}_{k,1}, \quad l_{\text{s}} = 1, \ldots, u^{\text{strip}} \end{aligned} \tag{5.32}$$

hold. As $X_{k,l_{\text{s}}} > 0$, we moreover observe that the phase equilibrium equations can be restated as

$$Y^{\text{rect}}_{k,l_{\text{r}}} = X^{\text{rect}}_{k,l_{\text{r}}} \frac{\sum_{j=1}^{k} \alpha_j \left(\frac{X^{\text{rect}}_{j,l_{\text{r}}}}{X^{\text{rect}}_{k,l_{\text{r}}}} - \frac{X^{\text{rect}}_{j-1,l_{\text{r}}}}{X^{\text{rect}}_{k,l_{\text{r}}}} \right)}{X^{\text{rect}}_{k,l_{\text{r}}} \sum_{j=1}^{k} \alpha_j \left(\frac{X^{\text{rect}}_{j,l_{\text{r}}}}{X^{\text{rect}}_{k,l_{\text{r}}}} - \frac{X^{\text{rect}}_{j-1,l_r}}{X^{\text{rect}}_{k,l_{\text{r}}}} \right) + \sum_{j=k+1}^{n} \alpha_j (X^{\text{rect}}_{j,l_{\text{r}}} - X^{\text{rect}}_{j-1,l_{\text{r}}})}. \tag{5.33}$$

Note further that the numerator in equation (5.33) is a convex combination of parameters $\alpha_1, \ldots, \alpha_k$. From $\alpha_k \leq \alpha_{k-1} \leq \cdots \leq \alpha_1$, it follows that the numerator is greater or equal to α_k. The denominator is a convex combination of the numerator and parameters $\alpha_{k+1}, \ldots, \alpha_n$. As $\alpha_n \leq \cdots \leq \alpha_{k+1} \leq \alpha_k$, we can conclude that the fractional term in the right-hand-side of equation (5.33) is greater or equal to one. This implies $Y^{\text{rect}}_{k,l_{\text{r}}} \geq X^{\text{rect}}_{k,l_{\text{r}}}$. In a similar way, we can verify that $Y^{\text{strip}}_{k,l_{\text{s}}} \geq X^{\text{strip}}_{k,l_{\text{s}}}$ holds. Combining these results, we get $\frac{\partial Y^{\text{rect}}_{k,l_{\text{r}}+1}}{\partial \nu_{\text{r}}} \leq 0$ and $\frac{\partial X^{\text{strip}}_{k,l_{\text{s}}+1}}{\partial \nu_{\text{s}}} \geq 0$.

Thus, given bounds on the arguments in (5.13) can be used to compute bounds on the aggregated concentration variables associated with the consecutive tray via standard interval arithmetic (e.g., see Hansen et al. [1991]). The resulting formulas are stated in the following two lemmas, where Lemma 5.6 addresses the stripping section and Lemma 5.7 deals with the rectifying section. In both lemmas, superscripts "strip" and "rect" are neglected in order to keep the notation simple.

Lemma 5.6. *(Stripping Section)*
Consider any $l_s \in \{1, \ldots, u\}$ and $k \in \{1, \ldots, n\}$. Assume further that ν_s ranges on $[\nu_s^{lo}, \nu_s^{up}]$, Y_{k,l_s} ranges on $[Y^{lo}_{k,l_s}, Y^{up}_{k,l_s}]$ and that $X_{k,1}$ ranges on $[X^{lo}_{k,1}, X^{up}_{k,1}]$. Then, lower and upper bounds X^{lo}_{k,l_s+1}, X^{up}_{k,l_s+1} on X_{k,l_s+1} are given by

$$X^{lo}_{k,l_s+1} = \nu_s^{lo} Y^{lo}_{k,l_s} + (1 - \nu_s^{lo}) X^{lo}_{k,1}, \quad X^{up}_{k,l_s+1} = \nu_s^{up} Y^{up}_{k,l_s} + (1 - \nu_s^{up}) X^{up}_{k,1}.$$

Lemma 5.7. *(Rectifying Section)*
Consider any $l_r \in \{1, \dots, u\}$ and $k \in \{1, \dots, n\}$. Assume further that ν_r ranges on $[\nu_r^{lo}, \nu_r^{up}]$, X_{k,l_r} ranges on $[X_{k,l_r}^{lo}, X_{k,l_r}^{up}]$ and that $Y_{k,1}$ ranges on $[Y_{k,1}^{lo}, Y_{k,1}^{up}]$. Then, lower and upper bounds Y_{k,l_r+1}^{lo}, Y_{k,l_r+1}^{up} on Y_{k,l_r+1} are given by

$$Y_{k,l_r+1}^{lo} = \nu_r^{up} X_{k,l_r}^{lo} + (1 - \nu_r^{up}) Y_{k,1}^{lo}, \quad Y_{k,l_r+1}^{up} = \nu_r^{lo} X_{k,l_r}^{up} + (1 - \nu_r^{lo}) Y_{k,1}^{up}.$$

Remark 5.8. It is worth to mention that the equations (5.32) do not hold for the variables used for the molar fractions in the original model formulation (Problem 5.1), e.g., see the molar fraction profile of the second component (blue-colored dashed curve) in Figure 5.2 (a). Thus, Lemma 5.6 and Lemma 5.7 are not applicable in the original formulation.

The purity constraints (5.18) already provide (strong) valid bounds on the concentration variables at the condenser and, hence, at the first tray of the rectifying section as well as on the concentration variables at the reboiler and the first tray (in our ordering) of the stripping section. Starting with these bounds, our next goal is to propagate bounds on the concentration variables tray by tray through each section by repeatedly applying the formulas for bound calculations from Lemma 5.6 and Lemma 5.7. This procedure defines the bound tightening strategy.

However, a re-use of the formulas from Lemma 5.6 and Lemma 5.7 will make it necessary to translate bounds on the aggregated concentration variables associated with the vapor or the liquid phase into valid bounds on the aggregated concentration variables in the respective other phase. This is achieved by exploiting the phase equilibrium equations (5.15) and their inverses (5.26), respectively, and leads to the formulas as given in Lemma 5.9 (for the stripping section) and in Lemma 5.10 (for the rectifying section). Again, superscripts "strip" and "rect" are omitted to keep the statements easy to read.

Lemma 5.9. ***(Stripping Section)***
Let $l_s \in \{1, \dots, u+1\}$ be fixed. Assume further that, for every $k = 1, \dots, n$, lower and upper bounds $X_{k,l_s}^{lo}, X_{k,l_s}^{up}$ on X_{k,l_s} are given, where

$$X_{k,l_s}^{lo} \le X_{k+1,l_s}^{lo} \quad \textit{and} \quad X_{k,l_s}^{up} \le X_{k+1,l_s}^{up} \quad \textit{hold for } k = 1, \dots, n-1.$$

Then, for each k, lower and upper bounds Y^{lo}_{k,l_s}*,* Y^{up}_{k,l_s} *on* Y_{k,l_s} *are given by*

$$Y^{lo}_{k,l_s} = \frac{\sum_{j=1}^{k} \alpha_j (X^{a_k}_{j,l_s} - X^{a_k}_{j-1,l_s})}{\sum_{j=1}^{n} \alpha_j (X^{a_k}_{j,l_s} - X^{a_k}_{j-1,l_s})} \quad \textit{and} \quad Y^{up}_{k,l_s} = \frac{\sum_{j=1}^{k} \alpha_j (X^{b_k}_{j,l_s} - X^{b_k}_{j-1,l_s})}{\sum_{j=1}^{n} \alpha_j (X^{b_k}_{j,l_s} - X^{b_k}_{j-1,l_s})},$$

where we define

$$X^{a_k}_{j,l_s} := \begin{cases} X^{lo}_{j,l_s}, & \textit{if } j \le k, \\ X^{up}_{j,l_s}, & \textit{if } j > k \end{cases} \qquad \textit{and} \qquad X^{b_k}_{j,l_s} = \begin{cases} X^{up}_{j,l_s}, & \textit{if } j \le k, \\ \max\{X^{up}_{k,l_s}, X^{lo}_{j,s}\}, & \textit{if } j > k \end{cases}$$

for $j = 1, \ldots, n$*.*

Proof. We interpret the aggregated phase equilibrium equations (5.15)

$$Y_{k,l_\mathrm{s}}(X) = \frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}{\sum_{j=1}^{n} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}$$

as functions in the liquid phase concentration variables. For all $k, q = 1, \ldots, n$ and for all $l_\mathrm{s} = 1, \ldots, u+1$, we consider the partial derivatives $\frac{\partial Y_{k,l_\mathrm{s}}(X)}{\partial X_{q,l_\mathrm{s}}}$ where we distinguish the three cases $q \le k-1$, $q = k$ and $q \ge k+1$. To keep the notation simple, we introduce the constant $\alpha_{n+1} := 0$.

For $q \le k-1$, we obtain

$$\begin{aligned} &\frac{\partial Y_{k,l_\mathrm{s}}(X)}{\partial X_{q,l_\mathrm{s}}} \\ &= \frac{(\alpha_q - \alpha_{q+1}) \sum_{j=1}^{n} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}{\big(\sum_{j=1}^{n} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})\big)^2} - \frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})(\alpha_q - \alpha_{q+1})}{\big(\sum_{j=1}^{n} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})\big)^2} \\ &= \frac{(\alpha_q - \alpha_{q+1}) \sum_{j=k+1}^{n} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}{\big(\sum_{j=1}^{n} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})\big)^2}. \end{aligned}$$

As $(\alpha_q - \alpha_{q+1}) \ge 0$ holds, this derivative is non-negative for all $k = 1, \ldots, n$ and all $l_\mathrm{s} = 1, \ldots, u+1$.

For $q = k$ we obtain

$$\begin{aligned} \frac{\partial Y_{q,l_\mathrm{s}}(X)}{\partial X_{q,l_\mathrm{s}}} &= \frac{\alpha_q \sum_{j=1}^{n} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}}) - \sum_{j=1}^{q} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})(\alpha_q - \alpha_{q+1})}{\big(\sum_{j=1}^{n} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})\big)^2} \\ &= \frac{\alpha_q \sum_{j=q+1}^{n} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}}) + \alpha_{q+1} \sum_{j=1}^{q} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}{\big(\sum_{j=1}^{n} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})\big)^2}, \end{aligned} \tag{5.34}$$

which is also non-negative for all $q = 1, \ldots, n$ and all $l_\mathrm{s} = 1, \ldots, u+1$.

For $q \geq k+1$ we obtain

$$\frac{\partial Y_{k,l_\mathrm{s}}(X)}{\partial X_{q,l_\mathrm{s}}} = \frac{-\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})(\alpha_q - \alpha_{q+1})}{\left(\sum_{j=1}^{n} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})\right)^2}, \tag{5.35}$$

which is non-positive for all $k = 1, \ldots, n$ and all $l_\mathrm{s} = 1, \ldots, u+1$ due to $(\alpha_q - \alpha_{q+1}) \geq 0$.

This shows that the phase equilibrium equations are component-wise monotonic. Therefore, we can apply simple interval arithmetic, again, leading to the following lower and upper bounds on the vapor phase concentration variables $Y_{k,l_\mathrm{s}+1}$.

$$Y^{\mathrm{lo}}_{k,l_\mathrm{s}} = \frac{\sum_{j=1}^{k} \alpha_j (X^{a_k}_{j,l_\mathrm{s}} - X^{a_k}_{j-1,l_\mathrm{s}})}{\sum_{j=1}^{n} \alpha_j (X^{a_k}_{j,l_\mathrm{s}} - X^{a_k}_{j-1,l_\mathrm{s}})} \quad \text{and} \quad Y^{\mathrm{up}}_{k,l_\mathrm{s}} = \frac{\sum_{j=1}^{k} \alpha_j (X^{b_k}_{j,l_\mathrm{s}} - X^{b_k}_{j-1,l_\mathrm{s}})}{\sum_{j=1}^{n} \alpha_j (X^{b_k}_{j,l_\mathrm{s}} - X^{b_k}_{j-1,l_\mathrm{s}})},$$

with

$$X^{a_k}_{j,l_\mathrm{s}} := \begin{cases} X^{\mathrm{lo}}_{j,l_\mathrm{s}}, & \text{if } j \leq k, \\ X^{\mathrm{up}}_{j,l_\mathrm{s}}, & \text{if } j > k \end{cases} \quad \text{and} \quad X^{b_k}_{j,l_\mathrm{s}} = \begin{cases} X^{\mathrm{up}}_{j,l_\mathrm{s}}, & \text{if } j \leq k, \\ X^{\mathrm{lo}}_{j,\mathrm{s}}, & \text{if } j > k. \end{cases} \tag{5.36}$$

for all $j = 1, \ldots, n$. We remark that the upper bound $Y^{\mathrm{up}}_{k,l_\mathrm{s}}$ on Y_{k,l_s} is not tight when $X^{\mathrm{up}}_{k,l_\mathrm{s}} > X^{\mathrm{lo}}_{k',l_\mathrm{s}}$ holds for some $k' > k$. In those cases, we can compute an improved upper bound on Y_{k,l_s} by finding the maximum of

$$Y_{k,l_\mathrm{s}}(X) = \frac{\sum_{j=1}^{k} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}{\sum_{j=1}^{n} \alpha_j (X_{j,l_\mathrm{s}} - X_{j-1,l_\mathrm{s}})}$$

restricted to $X^{\mathrm{lo}}_{k',l_\mathrm{s}} \leq X_{k,l_\mathrm{s}} \leq X_{k',l_\mathrm{s}} \leq X^{\mathrm{up}}_{k,l_\mathrm{s}}$. As $\frac{\partial Y_{k,l_\mathrm{s}}(X)}{\partial X_{k,l_\mathrm{s}}} \geq 0$ and $\frac{\partial Y_{k,l_\mathrm{s}}(X)}{\partial X_{k',l_\mathrm{s}}} \leq 0$ hold, it follows $X_{k,l_\mathrm{s}} = X_{k',l_\mathrm{s}}$ for the optimal solution. A comparison of the equations (5.34) and (5.35) gives rise to the following relation

$$\frac{\partial Y_{k,l_\mathrm{s}}(X)}{\partial X_{k,l_\mathrm{s}}} + \sum_{j=k+1}^{n} \frac{\partial Y_{k,l_\mathrm{s}}(X)}{\partial X_{j,l_\mathrm{s}}} \geq 0.$$

This shows that the maximum is attained at $X_{k,l_\mathrm{s}} = X_{k',l_\mathrm{s}} = X^{\mathrm{up}}_{k,l_\mathrm{s}}$. Hence, we can replace the definition of $X^{b_k}_{j,l_\mathrm{s}}$ in the equations (5.36) by

$$X^{b_k}_{j,l_\mathrm{s}} = \begin{cases} X^{\mathrm{up}}_{j,l_\mathrm{s}}, & \text{if } j \leq k, \\ \max\left\{X^{\mathrm{up}}_{k,\mathrm{s}}, X^{\mathrm{lo}}_{j,\mathrm{s}}\right\}, & \text{if } j > k. \end{cases}$$

This completes the proof. □

Lemma 5.10. ***(Rectifying Section)***
Let $l_r \in \{1, \dots, u+1\}$ be fixed. Assume further that, for every $k = 1, \dots, n$, lower and upper bounds Y^{lo}_{k,l_r}, Y^{up}_{k,l_r} on Y_{k,l_r} are given, where

$$Y^{lo}_{k,l_r} \leq Y^{lo}_{k+1,l_r} \quad \text{and} \quad Y^{up}_{k,l_r} \leq Y^{up}_{k+1,l_r} \quad \text{hold for } k = 1, \dots, n-1.$$

Then, for each k, lower and upper bounds X^{lo}_{k,l_r}, X^{up}_{k,l_r} on X_{k,l_r} are given by

$$X^{lo}_{k,l_r} = \frac{\sum_{j=1}^{k} \alpha_j^{-1}(Y^{a_k}_{j,l_r} - Y^{a_k}_{j-1,l_r})}{\sum_{j=1}^{n} \alpha_j^{-1}(Y^{a_k}_{j,l_r} - Y^{a_k}_{j-1,l_r})} \quad \text{and} \quad X^{up}_{k,l_r} = \frac{\sum_{j=1}^{k} \alpha_j^{-1}(Y^{b_k}_{j,l_r} - Y^{b_k}_{j-1,l_r})}{\sum_{j=1}^{n} \alpha_j^{-1}(Y^{b_k}_{j,l_r} - Y^{b_k}_{j-1,l_r})},$$

where we define

$$Y^{a_k}_{j,l_r} := \begin{cases} \min\{Y^{up}_{j,l_r}, Y^{lo}_{k,l_r}\}, & \text{if } j < k, \\ Y^{up}_{j,l_r}, & \text{if } j \geq k, \end{cases} \quad \text{and} \quad Y^{b_k}_{j,l_r} = \begin{cases} Y^{lo}_{j,l_r}, & \text{if } j < k, \\ Y^{up}_{j,r}, & \text{if } j \geq k \end{cases}$$

for $j = 1, \dots, n$.

Proof. Analog to the proof of Lemma 5.9. □

5.4.2 Domain Reduction using Fixed Points

Another way to tighten the variable bounds in the model formulation from Section 5.2 is to determine the fixed points from the recursive functions given by the component mass balance equations (5.13) and the phase equilibrium equations (5.15). Corollaries 5.4 and 5.5 imply that, for each aggregated component k, the four infinite sequences

$$\left\{X^{\text{strip}}_{k,l_s}\right\}_{l_s \in \mathbb{Z}_{\geq 1}}, \left\{Y^{\text{strip}}_{k,l_s}\right\}_{l_s \in \mathbb{Z}_{\geq 1}} \quad \text{and} \quad \left\{X^{\text{rect}}_{k,l_r}\right\}_{l_r \in \mathbb{Z}_{\geq 1}}, \left\{Y^{\text{rect}}_{k,l_r}\right\}_{l_r \in \mathbb{Z}_{\geq 1}}$$

must converge since they are monotonic and range on the bounded interval $[0, 1]$. Due to the monotonic behavior, the limit of each sequence further provides either a lower or an upper bound valid for each element of the sequence. As done in [Kunde et al., 2016] for the two-component case, we can exploit this property by incorporating, for each $k = 1, \dots, n$, the following (redundant) nonlinear constraints to the aggregated model formulation.

$$
\begin{aligned}
X_k^{\text{strip},\star} &= \nu_{\text{s}} Y_k^{\text{strip},\star} + (1-\nu_{\text{s}}) X_{k,1}^{\text{strip}}, \\
Y_k^{\text{strip},\star} &= \frac{\sum_{j=1}^{k} \alpha_j (X_j^{\text{strip},\star} - X_{j-1}^{\text{strip},\star})}{\sum_{j=1}^{n} \alpha_j (X_j^{\text{strip},\star} - X_{j-1}^{\text{strip},\star})}, \\
X_k^{\text{strip},\star} &\geq X_{k,l_{\text{s}}}^{\text{strip}}, \quad Y_k^{\text{strip},\star} \geq Y_{k,l_{\text{s}}}^{\text{strip}}, \qquad l_{\text{s}} = 1,\dots,u^{\text{strip}}+1, \\
Y_k^{\text{rect},\star} &= \nu_{\text{r}} X_k^{\text{rect},\star} + (1-\nu_{\text{r}}) Y_{k,1}^{\text{rect}}, \\
X_k^{\text{rect},\star} &= \frac{\sum_{j=1}^{k} \alpha_j^{-1} (Y_j^{\text{rect},\star} - Y_{j-1}^{\text{rect},\star})}{\sum_{j=1}^{n} \alpha_j^{-1} (Y_j^{\text{rect},\star} - Y_{j-1}^{\text{rect},\star})}, \\
X_k^{\text{rect},\star} &\leq X_{k,l_{\text{r}}}^{\text{rect}}, \quad Y_k^{\text{rect},\star} \leq Y_{k,l_{\text{r}}}^{\text{rect}}, \qquad l_{\text{r}} = 1,\dots,u^{\text{rect}}+1.
\end{aligned} \tag{5.37}
$$

Adding these fixed point equations to the model formulation can be interpreted as an alternative to the bound tightening strategy. Both approaches are motivated by the monotonic behavior of the concentration profiles. In general, the fixed point equations lead to weaker bounds than the bounds that can be obtained by the bound tightening strategy, as the latter ones are monotonic with respect to the section lengths and also bounded by the fixed points.

However, it may be useful to integrate the fixed point equations into the model formulation instead of using the bound tightening strategy. This may be assumed, as the bound tightening strategy has to be explicitly applied at every node of the Branch and Bound tree, while the fixed point equations are integrated by simply adding several equations to the model formulation. The computational study in Section 5.5 is also designed to investigate this difference. Furthermore, applying the bound tightening strategy requires the user to modify the solution process which is not always possible for commercial software.

5.5 Computational Results

In this section, we computationally evaluate the impact of the presented techniques on the performance of global optimization software. For this, we consider several numerical test instances dealing with ideal multi-component distillation processes. The objective of all instances is to find an optimal column design w.r.t. cost function (5.19), that separates the more volatile components from the less volatile components up to a given purity.

5.5.1 Test Setting

We consider 16 test instances. The reference test instance *ref* consists of a mixture of $n = 4$ components with initial composition $x_i^{\text{in}} = \frac{1}{4}$ for $i = 1, 2, 3, 4$, and with the constant relative volatilities $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (6, 4, 1.2, 1)$. The number l^{col} of trays that can be used in the entire distillation column is bounded by 25. Every section consists of at least one tray, so that the upper bound on the number of trays used for the rectifying as well as for the stripping section is given by $u^{\text{rect}} = u^{\text{strip}} = 23$. Molar flows F, B, D and V are given in terms of $\text{mol}\,\text{s}^{-1}$. The feed molar flow F is fixed to 1, while the remaining molar flows are variable and may range as follows: $0 \text{ mol}\, s^{-1} \leq V \leq 20 \text{ mol}\, s^{-1}$, $0 \text{ mol}\, s^{-1} \leq B, D \leq 1 \text{ mol}\, s^{-1}$. We choose the split σ to be 2. Recall from Section 3 that σ defines the more volatile single components $(1, \ldots, \sigma)$ withdrawn from the condenser and the less volatile single components $(\sigma + 1, \ldots, n)$ withdrawn from the reboiler. With respect to split σ, we call the components σ and $\sigma + 1$ *key components*, while the others are called *non-key components.* The purity requirements are given by $\pi^{\text{dist}} = \pi^{\text{bot}} = 0.99$.

The remaining test instances are defined by changing the values of several parameters, resulting in five groups of further test instances that are briefly explained next.

The first group is defined by varying the constant relative volatilities for the non-key components from the reference instance. The specifications are given in Table 5.3.

Instance	*adis1*	*adis2*	*adis3*
$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$	$(12, 4, 1.2, 1)$	$(12, 8, 2.4, 1)$	$(24, 8, 2.4, 1)$

Table 5.3: Specification of test instances with change in the volatilities of non-key components

The second group consists of two further instances for which the split σ is changed. Moreover, the constant relative volatilities are adapted in such a way that the ratios between the volatilities of the key components are the same as in the reference setting. Table 5.4 shows the concrete specifications.

Instance	*apos1*	*apos2*
σ	1	3
$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$	$(6.67, 2, 1.5, 1)$	$(8, 5, 3.33, 1)$

Table 5.4: Specification of test instances with change in the split σ

Instance	*con1*	*con2*	*con3*
$(x_1^{\text{in}}, x_2^{\text{in}}, x_3^{\text{in}}, x_4^{\text{in}})$	$(\frac{1}{10}, \frac{2}{5}, \frac{2}{5}, \frac{1}{10})$	$(\frac{1}{4}, \frac{2}{5}, \frac{1}{10}, \frac{1}{4})$	$(\frac{1}{4}, \frac{1}{10}, \frac{2}{5}, \frac{1}{4})$

Table 5.5: Specification of test instances given by varying the initial composition of the mixture

In the third group, we change the initial composition of the mixture as given in Table 5.5.

Group four consists of the test instances for which we vary the purity requirements on condenser and reboiler. In addition, we adapt the constant relative volatilities in order to keep the separation processes approximately as difficult as the separation process of the reference instance. Table 5.6 provides the specific setting for the changed parameters.

Instance	*pur1*	*pur2*	*pur3*
$(\pi^{\text{dist}}, \pi^{\text{bot}})$	$(0.95, 0.95)$	$(0.99, 0.95)$	$(0.95, 0.99)$
$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$	$(3.64, 2.42, 1.2, 1)$	$(4.62, 3.08, 1.2, 1)$	$(4.62, 3.08, 1.2, 1)$

Table 5.6: Specification of test instances with different purity requirements

Finally, we define a fifth group of test instances in which different numbers of components are considered. For each such instance, we adapt split σ, initial composition and relative volatilities accordingly, as summarized in Table 5.7.

5.5.2 Problem Formulation

For each instance, two MINLP formulations are derived. The first formulation, called *MINLP-O*, is based on the original distillation column model (Problem 5.1) as presented in Section 5.1. The second formulation makes use of the

Instance	*comp1*	*comp2*	*comp3*	*comp4*
n	2	3	5	5
σ	1	1	2	1
$(x_1^{\text{in}}, \dots, x_n^{\text{in}})$	$(\frac{1}{2}, \frac{1}{2})$	$(\frac{1}{2}, \frac{1}{4}, \frac{1}{4})$	$(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$	$(\frac{1}{2}, \frac{1}{4}, \frac{1}{10}, \frac{3}{40}, \frac{3}{40})$
$(\alpha_1, \dots, \alpha_n)$	$(3.33, 1)$	$(4, 1.2, 1)$	$(6, 4, 1.2, 1.1, 1)$	$(6.67, 2, 1.5, 1.2, 1)$

Table 5.7: Test instances where the number of components is changed

reformulated model (Problem 5.2) with aggregated concentration variables as introduced in Section 5.2 and is called *MINLP-A*. Additionally, we apply several different solution strategies to *MINLP-A*. The first strategy, indicated by *w/Mo*, adds the (redundant) conditions on monotonicity from Corollaries 5.4 and 5.5. Note that these conditions are in general not equivalent to the bound tightening strategy. This holds as the bound tightening additionally relies on the detection of monotonicity in the constraints and specialized interval arithmetic. The second strategy takes the fixed-point equations (5.37) into account and is labeled by *w/Fix*.

All formulations have been implemented using the following standard reformulation techniques. Due to their redundancy, all variables that are associated with the last component n, as well as the variables $x_i^{\text{dist}}, x_i^{\text{bot}}$, $i = 1, \dots, n$, and $X_k^{\text{dist}}, X_k^{\text{bot}}$, $k = 1, \dots, n$ are eliminated. Moreover, each constraint containing a rational function is multiplied by its denominator and restated as a polynomial constraint.

Note that the coupling conditions (5.6a) and (5.17a) involve quadratic terms including binary variables. We use a standard approach to linearize these types of equations. We applied several different linearization techniques and found, based on preliminary computations, that the following one is best suited for our cases. Consider the first line of the equations (5.17a)

$$X_k^{\text{feed}-1} = \sum_{l_{\text{r}}=1}^{u^{\text{rect}}} \beta_{l_{\text{r}}}^{\text{rect}} X_{k,l_{\text{r}}}^{\text{rect}},$$

for a fixed $k \in \{1, \dots, n\}$. Using that $\sum_{l_{\text{r}}=1}^{u^{\text{rect}}} \beta_{l_{\text{r}}}^{\text{rect}} = 1$ holds, we can reformulate the equation as

$$\beta_{l_{\mathrm{r}}}^{\mathrm{rect}} X_k^{\mathrm{feed}-1} = \beta_{l_{\mathrm{r}}}^{\mathrm{rect}} X_{k,l_{\mathrm{r}}}^{\mathrm{rect}}, \qquad l_{\mathrm{r}} = 1, \dots, u^{\mathrm{rect}}.$$

Next, we introduce a new variable $C_{l_{\mathrm{r}}}$ for $l_{\mathrm{r}} = 1, \dots, u^{\mathrm{rect}}$ and demand

$$C_{l_{\mathrm{r}}} = \beta_{l_{\mathrm{r}}}^{\mathrm{rect}} X_k^{\mathrm{feed}-1}, \quad C_{l_{\mathrm{r}}} = \beta_{l_{\mathrm{r}}}^{\mathrm{rect}} X_{k,l_{\mathrm{r}}}^{\mathrm{rect}}, \qquad l_{\mathrm{r}} = 1, \dots, u^{\mathrm{rect}}.$$

For every $l_{\mathrm{r}} = 1, \dots, u^{\mathrm{rect}}$, these two quadratic equations are linearized in the following well-known way.

$$\begin{aligned} &C_{l_{\mathrm{r}}} \geq \beta_{l_{\mathrm{r}}}^{\mathrm{rect}} + X_k^{\mathrm{feed}-1} - 1, \qquad &&C_{l_{\mathrm{r}}} \geq \beta_{l_{\mathrm{r}}}^{\mathrm{rect}} + X_{k,l_{\mathrm{r}}}^{\mathrm{rect}} - 1, && \\ &C_{l_{\mathrm{r}}} \leq \beta_{l_{\mathrm{r}}}^{\mathrm{rect}}, &&C_{l_{\mathrm{r}}} \leq X_k^{\mathrm{feed}-1}, \qquad &&C_{l_{\mathrm{r}}} \leq X_{k,l_{\mathrm{r}}}^{\mathrm{rect}}. \end{aligned}$$

The remaining equations in (5.17a) and (5.6a) are handled analogously.

All computations are carried out on a 3.00 GHz Intel Xeon E5450 Processor with a limit of 30 GB memory space for each run. Moreover, running time is limited to 24 hours and the relative optimality gap is chosen to be 10^{-4}.

In order to compare different solution strategies for our test instances, we utilize a standard performance measure. It is given by the geometric mean of the solution times for each instance relative to a reference strategy. We will use this as an auxiliary tool in our analysis. For calculating the average solution time, we use 24 hours for all instances that are not solved within the time limit. However, we will also display and discuss individual results, as the number of test instances is quite restricted.

5.5.3 Results using SCIP

All MINLPs are implemented and solved with SCIP 3.2 (Achterberg [2009]) using CPLEX 12.6.0 (IBM CPLEX [2014]) as LP-subsolver and IPOPT 3.12.4 (Wächter and Biegler [2006]) (incl. HSL-routines MA27 and MC19 (HSL)) as NLP-subsolver.

By using SCIP we are able to apply a third solution strategy to *MINLP-A*. Therein, the bound tightening strategy as described in Section 5.4 is used at every node in the Branch and Bound tree. This is achieved by implementing a domain propagation routine as an own `constraint_handler` in SCIP. The label *w/BT* indicates that the bound tightening strategy is switched on.

Our computational results are summarized in the Tables 5.8 to 5.11. Table 5.8 shows the running time in CPU minutes for all instances that have been solved in the time limit of 24 hours. Table 5.9 displays the total number of Branch and Bound nodes needed in the solution process for these instances. Table 5.10 lists the relative gap between upper and lower bound in percentage after 24 hours for all instances that have not been solved in this time. Table 5.11 shows the geometric mean of the solution times for each instance relative to the standard formulation *MINLP-O* and to other formulations, respectively.

First we compare the original formulation (*MINLP-O*) to the aggregated one (*MINLP-A*). By using formulation *MINLP-O*, eight of the sixteen instances are not solved to global optimality within the time limit (see column 2 of Table 5.8). The same holds when formulation *MINLP-A* is used, but with a different subset of unsolved instances (see column 3 of Table 5.8). Comparing the two columns, we see that among all instances that are solved by both formulations, using formulation *MINLP-A* leads to a lower running time for all but instance *comp2*, and to a lower number of Branch and Bound nodes needed for all these instances (see columns 2, 3 of Table 5.9). Among all instances that are not solved by both formulations, the remaining optimality gap is significantly lower when formulation *MINLP-A* is used (see columns 2, 3 of Table 5.10). On average, the running time is reduced to 51.3% as shown in row 2 of Table 5.11. These observations suggest an advantageous behavior of formulation *MINLP-A* during the solution process. However, we need to mention that applying the reformulation alone does not always lead to an improvement, since three of our test instances are solved by using formulation *MINLP-O* but not by using formulation *MINLP-A*.

Next, we discuss the influence of adding the redundant monotonicity constraints (*w/Mo*) to our model formulations by examining the differences between columns 3 and 4 of Tables 5.8 and 5.9. For five of our test instances, the influence is negative for both running time and number of nodes needed in the solution process, while the opposite holds for ten instances. Only instance *apos1* is not solved by formulation *MINLP-A w/Mo*, but still the remaining

Ex.	*MINLP-O*	*MINLP-A*	*MINLP-A*	*MINLP-A*	*MINLP-A*	*MINLP-A*
			w/Mo	*w/Mo,Fix*	*w/BT*	*w/Mo,BT*
ref	–	95	61	90	9	**6**
adis1	280	–	88	123	24	**13**
adis2	–	–	372	122	26	**8**
adis3	–	–	34	40	13	**12**
apos1	249	–	–	92	15	**12**
apos2	–	51	34	20	5	**4**
con1	232	15	30	74	6	**5**
con2	–	–	36	127	**25**	45
con3	1038	97	134	–	19	**10**
pur1	–	–	658	405	64	**32**
pur2	–	–	1331	110	21	**12**
pur3	1373	462	71	–	57	**13**
comp1	1.1	0.4	0.5	0.4	0.2	**0.1**
comp2	12	14	33	28	3	**1**
comp3	–	157	290	–	**41**	46
comp4	481	–	443	–	32	**20**

Table 5.8: Running time in CPU minutes using the SCIP framework. Label "–" means that the problem is not solved within the time limit of 24 hours. The lowest running time for every instance is highlighted.

Ex.	*MINLP-O*	*MINLP-A*	*MINLP-A*	*MINLP-A*	*MINLP-A*	*MINLP-A*
			w/Mo	*w/Mo,Fix*	*w/BT*	*w/Mo,BT*
ref	–	148	100	119	10	**7**
adis1	393	–	118	153	31	**19**
adis2	–	–	558	217	31	**9**
adis3	–	–	61	64	**16**	16
apos1	392	–	–	123	19	**17**
apos2	–	91	53	34	**5**	5
con1	346	19	47	95	6	**5**
con2	–	–	61	180	**31**	47
con3	1539	125	216	–	21	**12**
pur1	–	–	920	514	60	**35**
pur2	–	–	2285	151	24	**13**
pur3	1526	623	110	–	55	**14**
comp1	18	4	5	4	1	**1**
comp2	49	48	133	96	8	**5**
comp3	–	111	219	–	28	**24**
comp4	327	–	417	–	22	**16**

Table 5.9: Branch and Bound nodes (in 1000) needed for solving the problem using the SCIP framework. Label "–" means that the problem is not solved within the time limit of 24 hours. The lowest number of nodes needed for every instance is highlighted.

Ex.	*MINLP-O*	*MINLP-A*	*MINLP-A*	*MINLP-A*	*MINLP-A*	*MINLP-A*
			w/Mo	*w/Mo,Fix*	*w/BT*	*w/Mo,BT*
ref	18.36	–	–	–	–	–
adis1	–	0.33	–	–	–	–
adis2	13.02	8.15	–	–	–	–
adis3	13.44	0.26	–	–	–	–
apos1	–	0.27	0.02	–	–	–
apos2	18.25	–	–	–	–	–
con1	–	–	–	–	–	–
con2	6.11	0.11	–	–	–	–
con3	–	–	–	0.33	–	–
pur1	11.07	0.10	–	–	–	–
pur2	20.97	7.9	–	–	–	–
pur3	–	–	–	1.17	–	–
comp1	–	–	–	–	–	–
comp2	–	–	–	–	–	–
comp3	22.62	–	–	0.03	–	–
comp4	–	0.02	–	0.03	–	–

Table 5.10: Relative gap given in percentage after 24 hours using the SCIP framework. Label “–” means that the problem was solved with a gap lower than 0.01%.

relative to	*MINLP-O*	*MINLP-A*	*MINLP-A w/Mo*	*MINLP-A w/Mo,Fix*	*MINLP-A w/BT*	*MINLP-A w/Mo,BT*
-O	100%	51.3%	21.7%	27.1%	2.9%	**1.9%**
-A	–	100%	42.3%	–	5.6%	–
w/BT	–	–	–	–	100%	66.2%
w/Mo	–	–	100%	125.0%	–	8.8%

Table 5.11: Geometric mean of the running times relative to selected reference formulations using the SCIP framework.

optimality gap is lower than the one obtained by using formulation *MINLP-A*. The average running time is reduced to 42% by using *MINLP-A w/Mo* instead of *MINLP-A* (row 3 in Table 5.11). Thus we can conclude that the positive influence of the additional constraints for a wide subset of our instances dominates the negative influences. A similar result can be obtained by comparing the columns 6 and 7 of Tables 5.8 and 5.9. All but two instances have a lower running time and all but three instances have a lower amount of nodes needed when the additional monotonicity constraints are added. The average running time is reduced to 66% in this case (row 4 in Table 5.11).

Now we analyze the influence of the fixed point equations (*w/Fix*). For this, we consider the columns 3 and 4 in Table 5.8. For two instances (*apos1, pur2*), this influence is significantly positive while in four other cases (*con3, pur3, comp3, comp4*), the respective instances could not be solved in the time limit. For all other instances, the influence on running time and nodes needed (Table 5.9) is very mixed and differences are not as significant. The average running time by adding the fixed point strategy is increased to 125% (row 5 in Table 5.11). We conclude an overall small, but rather negative influence of this solution strategy, with a huge impact in some special cases.

At last we focus on our main contribution in terms of algorithmic impact, which is the problem specific bound tightening strategy (*w/BT*). Note that the two solution strategies using this method (columns 6 and 7 in Tables 5.8 and 5.9) are the only ones able to solve all our instances to global optimality during the given time limit. Furthermore, one of these two strategies is always the best in terms of both, running time and Branch and Bound nodes needed. By comparing the columns 3 and 6 in Table 5.8, we can see a huge improvement in running time by applying the bound tightening strategy to the aggregated model formulation. On average, the running time is reduced to 5.6% (row 3 of Table 5.11). A similar result is obtained by analyzing the influence of the bound tightening on the model formulation already using the monotonicity constraints (columns 4 and 7 in Table 5.8). The average running time in this case is reduced to 8.8% (row 5 of Table 5.11). These results show a significant performance improvement by applying our developed bound tightening strategy during the optimization process of ideal multi-component distillation columns.

To summarize the analysis, three of our four proposed solution strategies have a positive influence on the performance of SCIP on our test set. These strategies are the aggregated reformulation, the monotonicity constraints and especially the bound tightening. If we add all three strategies and compare them to the original model formulation (columns 2 and 7 in Table 5.8), we can derive an average reduction of the running time to 1.9% (row 2 of Table 5.11).

5.5.4 Results using BARON

We finally investigate the computational behavior of another global optimization solver on our MINLP formulations. For this, we chose the standard solver BARON 16.3.4 (Tawarmalani and Sahinidis [2005]) as provided within the modeling system GAMS 24.7.1 (GAMS Development Corporation [2016]). The solver is used with default settings, CPLEX as LP-subsolver and CONOPT as NLP-subsolver. We remark that our focus is on the question how the solver works as a black box on the different model formulations rather than on comparing the performance with SCIP.

In the following, we consider the model formulations *MINLP-O*, *MINLP-A*, *MINLP-A w/Mo* and *MINLP-A w/Mo,Fix* as defined in Subsection 6.2. We do not see a way to implement the bound tightening strategy in the closed-source environment GAMS, so that this strategy is excluded from further considerations.

Table 5.12 displays the running time in CPU minutes and Table 5.13 displays the number of Branch and Bound nodes needed for the solution process. The single instance that has not been solved during the time limit using formu-

Instance	*MINLP-O*	*MINLP-A*	*MINLP-A*	*MINLP-A*
			w/Mo	*w/Mo,Fix*
ref	240	201	**51**	80
adis1	**46**	103	59	95
adis2	–	307	**131**	160
adis3	110	220	114	**103**
apos1	46	68	**36**	53
apos2	131	**29**	83	41
con1	117	55	**41**	242
con2	43	185	**23**	34
con3	232	174	163	**121**
pur1	376	559	88	**78**
pur2	332	**126**	190	178
pur3	392	172	**36**	53
comp1	0.8	0.7	0.7	**0.4**
comp2	**4**	14	22	15
comp3	596	**156**	304	234
comp4	**56**	182	355	590

Table 5.12: Running time in CPU minutes using GAMS:BARON. Label "–" means that the problem is not solved within the time limit of 24 hours. The lowest running time for every instance is highlighted.

Instance	*MINLP-O*	*MINLP-A*	*MINLP-A w/Mo*	*MINLP-A w/Mo,Fix*
ref	47	11	**6**	7
adis1	20	13	**12**	13
adis2	–	37	28	**24**
adis3	22	11	**5**	13
apos1	9	7	**7**	8
apos2	3	**0.2**	14	3
con1	24	15	**8**	19
con2	18	30	**3**	5
con3	53	16	28	**5**
pur1	25	14	5	**1**
pur2	35	**7**	19	17
pur3	46	3	5	**0.3**
comp1	1	0.3	2	**0.1**
comp2	3	**3**	17	5
comp3	18	**5**	9	8
comp4	12	**9**	17	12

Table 5.13: Branch and Bound nodes (in 1000) needed for solving the problem using GAMS:BARON. Label "–" means that the problem is not solved within the time limit of 24 hours. The lowest number of nodes needed for every instance is highlighted.

relative to	*MINLP-O*	*MINLP-A*	*MINLP-A w/Mo*	*MINLP-A w/Mo,Fix*
-O	100%	91.4%	**60.4%**	69.2%
-A	–	100%	66.1%	–
w/Mo	–	–	100%	114.5%

Table 5.14: Geometric mean of the running times relative to selected reference formulations using GAMS:BARON.

lation *MINLP-O* is indicated by the symbol "–" and has a remaining optimality gap of $19,46\%$ after 24 hours. Table 5.14 shows the geometric mean of the running times relative to *MINLP-O*, *MINLP-A* and *MINLP-A w/Mo* respectively. Again we will analyze the influence of the different formulations and solution strategies one by one.

Using the reformulation *MINLP-A* instead of *MINLP-O* has an ambiguous influence on the solution time of our test instances. Nine of the sixteen instances are solved faster while the other seven are solved slower (columns 2 and 3 in Table 5.12). On average, the running time is reduced to 91.4% as shown in row 2 of Table 5.14. However, it is important to note that all instances are solved by using *MINLP-A*, and that all but one instance needed a lower amount of Branch and Bound nodes.

Next we analyze the effect of adding the monotonicity constraints (*w/Mo*) to our model formulation (columns 3 and 4 in Table 5.12). Although there are six instances with a higher running time using this solution strategy, the overall influence is very positive. On average, the running time is reduced to 66% (row 3 in Table 5.14)

Further, adding the fixed point equations (*w/Mo,Fix*) to this formulation has a small negative influence on the performance of the solver on our test set. Half of the instances perform better and the other half performs worse in terms on running time (columns 4 and 5 in Table 5.14). On average, the running time is increased to 114.5% (row 4 in Table 5.14).

The differences between the performances of our problem formulations are considerably smaller using BARON as a solver instead of SCIP. Nevertheless, we can detect the same general tendencies. The aggregated formulation has a small positive influence and the monotonicity constraints a significantly higher one. The fixed point equations tend to reduce the performance of the solvers. Unfortunately, we are not able to compare the influence of the bound tightening strategy on the solver BARON.

We observe that some of our instances benefit a lot from our solution strategies while others are rather disturbed. We assume that this holds for the following reason. Adding additional constraints on the one hand tightens the model formulation, but on the other hand increases the problem size. The respective trade off in terms of solver performance varies among the instances and leads to the observed behavior.

Chapter 6

Conclusion

In this thesis, we presented general theoretical results for MINLPs as well as specialized techniques for applications in engineering. In particular, we discussed the role of the convex relaxation of the feasible set for the spatial Branch and Bound algorithm. We developed several relaxation refinement strategies and illustrated their positive influence on the solution process by two computational studies.

Solving MINLPs with a large number of variables and constraints, in particular those arising from applications, often requires an unreasonable computational effort. Our results confirm the obvious assumption, that the quality of the convex relaxation has a crucial impact on this required effort. However, significant improvements for the relaxation of general MINLPs are rarely expected. Our theoretical result allows for a quite general cutting plane approach, but has relatively high requirements concerning the applicability. Therefore, it is often worthwhile to consider special problem classes or applications. We showed that the computational effort could be reduced in our cases by analyzing the underlying constraint set and by adapting existing optimization techniques.

Despite recent progress and a broad collection of literature, there is still room for improvement in this context. This holds for the general theory as well as for problem specific methods. The required progress is usually performed in small steps and relies on the combination of multiple results. We hope that this work contributes to his part in order to fill the remaining gap.

Bibliography

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

Kurt M. Anstreicher and Samuel Burer. Computable representations for convex hulls of low-dimensional quadratic forms. *Mathematical Programming*, 124(1):33–43, 2010. doi:10.1007/s10107-010-0355-9.

Martin Ballerstein. *Convex Relaxations for Mixed-Integer Nonlinear Programs*. PhD thesis, Eidgenössische Technische Hochschule Zürich, 2013.

Martin Ballerstein, Dennis Michaels, and Stefan Vigerske. Linear underestimators for bivariate functions with a fixed convexity behavior. Technical Report 13-02, Zuse Institute Berlin, 2013.

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. *Optimization and Engineering*, 16(2):409–440, 2015. doi:10.1007/s11081-014-9267-5.

Pietro Belotti, Andrew J. Miller, and Mahdi Namazifar. Valid Inequalities and Convex Hulls for Multilinear Functions. *Electronic Notes in Discrete Mathematics*, 36:805 – 812, 2010. doi:10.1016/j.endm.2010.05.102.

Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013. doi:10.1017/S0962492913000032.

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

Samuel Burer and Adam N. Letchford. Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97 – 106, 2012. doi:10.1016/j.sorms.2012.08.001.

Samuel Burer and Yinyu Ye. Exact semidefinite formulations for a class of (random and non-random) nonconvex quadratic programs. *Mathematical Programming*, 2019. doi:10.1007/s10107-019-01367-2.

Michael R. Bussieck and Stefan Vigerske. MINLP Solver Software. In *Wiley Encyclopedia of Operations Research and Management Science*. American Cancer Society, 2011. doi:10.1002/9780470400531.eorms0527.

James E Falk. Lagrange multipliers and nonconvex programs. *SIAM Journal on Control*, 7(4):534–545, 1969.

GAMS Development Corporation. General Algebraic Modeling System (GAMS) Release 24.7.1. Washington, DC, USA, 2016. URL `http://www.gams.com/`.

Björn Geißler, Alexander Martin, Antonio Morsi, and Lars Schewe. Using Piecewise Linear Functions for Solving MINLPs. In Jon Lee and Sven Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 287–314, New York, NY, 2012. Springer New York. ISBN 978-1-4614-1927-3.

Ambros Gleixner, Leon Eifler, Tristan Gally, Gerald Gamrath, Patrick Gemander, Robert Lion Gottwald, Gregor Hendel, Christopher Hojny, Thorsten Koch, Matthias Miltenberger, Benjamin Müller, Marc E. Pfetsch, Christian Puchert, Daniel Rehfeldt, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Jan Merlin Viernickel, Stefan Vigerske, Dieter Weninger, Jonas T. Witt, and Jakob Witzig. The SCIP Optimization Suite 5.0. Technical Report 17-61, Zuse Institute Berlin, 2017.

Ambros M. Gleixner, Timo Berthold, Benjamin Müller, and Stefan Weltge. Three enhancements for optimization-based bound tightening. *Journal of Global Optimization*, pages 1–27, 2016. doi:10.1007/s10898-016-0450-4.

Ignacio E. Grossmann, Jose Antonio Caballero, and Hector Yeomans. Mathematical Programming Approaches to the Synthesis of Chemical Process Systems. *Korean Journal of Chemical Engineering*, 16(4):407–426, 1999. doi:10.1007/BF02698263.

Pierre Hansen, Brigitte Jaumard, and Shi-Hui Lu. An analytical approach to global optimization. *Mathematical Programming*, 52(1-3):227–254, 1991. doi:10.1007/BF01582889.

HSL. A collection of Fortran codes for large-scale scientific computation. URL `http://www.hsl.rl.ac.uk/`.

IBM CPLEX, 2014. URL `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`.

Matthias Jach, Dennis Michaels, and Robert Weismantel. The Convex Envelope of $(n-1)$-Convex Functions. *SIAM Journal on Optimization*, 19(3):1451–1466, 2008. doi:10.1137/07069359X.

Ravindran Kannan and Clyde L. Monma. On the Computational Complexity of Integer Programming Problems. In Rudolf Henn, Bernhard Korte, and Werner Oettli, editors, *Optimization and Operations Research*, pages 161–172. Springer Berlin Heidelberg, 1978. ISBN 978-3-642-95322-4.

James E Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

Klaus Kleibohm. Bemerkungen zum Problem der nichtkonvexen Programmierung. *Unternehmensforschung*, 11(1):49–60, 1967. doi:10.1007/BF01922383.

Thorsten Koch, Benjamin Hiller, Marc Pfetsch, and Lars Schewe, editors. *Evaluating Gas Network Capacities.* Society for Industrial and Applied Mathematics, 2015. doi:10.1137/1.9781611973693.

Christian Kunde, Dennis Michaels, Jovana Micovic, Philip Lutze, Andrzej Górak, and Achim Kienle. Deterministic global optimization in conceptual process design of distillation and melt crystallization. *Chemical Engineering and Processing: Process Intensification*, 99:132 – 142, 2016. doi:10.1016/j.cep.2015.09.010.

Claude Lemaréchal. Chapter VII Nondifferentiable Optimization. In *Optimization*, volume 1 of *Handbooks in Operations Research and Management Science*, pages 529 – 572. Elsevier, 1989. doi:10.1016/S0927-0507(89)01008-X.

Marco Locatelli and Fabio Schoen. *Global Optimization: Theory, Algorithms, and Applications.* Society for Industrial and Applied Mathematics, 2013. doi:10.1137/1.9781611972672.

Marco Locatelli and Fabio Schoen. On convex envelopes for bivariate functions over polytopes. *Mathematical Programming*, 144(1):65–91, 2014. doi:10.1007/s10107-012-0616-x.

Marko Mäkelä. Survey of Bundle Methods for Nonsmooth Optimization. *Optimization Methods and Software*, 17(1):1–29, 2002. doi:10.1080/10556780290027828.

Alexander Martin, Markus Möller, and Susanne Moritz. Mixed Integer Models for the Stationary Case of Gas Network Optimization. *Mathematical Programming*, 105(2):563–582, 2006. doi:10.1007/s10107-005-0665-5.

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

Maximilian Merkert. *Solving Mixed-Integer Linear and Nonlinear Network Optimization Problems by Local Reformulations and Relaxations.* PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2017.

Alfons Mersmann, Matthias Kind, and Johann Stichlmair. *Thermal Separation Technology.* Springer Heidelberg Dordrecht London New York, 2011.

Nick Mertens, Christian Kunde, Achim Kienle, and Dennis Michaels. Monotonic reformulation and bound tightening for global optimization of ideal multi-component distillation columns. *Optimization and Engineering*, 19 (2):479–514, 2018. doi:10.1007/s11081-018-9377-6.

Clifford A. Meyer and Christodoulos A. Floudas. Convex envelopes for edge-concave functions. *Mathematical Programming*, 103(2):207–224, 2005. doi:10.1007/s10107-005-0580-9.

Ruth Misener and Christodoulos A. Floudas. ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. *Journal of Global Optimization*, 2014. doi:10.1007/s10898-014-0166-2.

Ulaganathan Nallasivam, Vishesh H. Shah, Anirudh A. Shenvi, Joshua Huff, Mohit Tawarmalani, and Rakesh Agrawal. Global optimization of multicomponent distillation configurations: 2. Enumeration based global minimization algorithm. *AIChE Journal*, 62(6):2071–2086, 2016. doi:10.1002/aic.15204.

Marc E. Pfetsch, Armin Fügenschuh, Björn Geißler, Nina Geißler, Ralf Gollmer, Benjamin Hiller, Jesco Humpola, Thorsten Koch, Thomas Lehmann, Alexander Martin, Antonio Morsi, Jessica Rövekamp, Lars Schewe, Martin Schmidt, Rüdiger Schultz, Robert Schwarz, Jonas Schweiger, Claudia Stangl, Marc C. Steinbach, Stefan Vigerske, and Bernhard M. Willert. Validation of Nominations in Gas Network Optimization: Models, Methods, and Solutions. *Optimization Methods and Software*, 30(1): 15–53, 2015. doi:10.1080/10556788.2014.888426.

Ignacio Quesada and Ignacio E. Grossmann. Global Optimization Algorithm for Heat Exchanger Networks. *Industrial & Engineering Chemistry Research*, 32(3):487–499, 1993. doi:10.1021/ie00015a012.

Helmut Ratschek and Jon Rokne. Interval Methods. In Reiner Horst and Panos M. Pardalos, editors, *Hand of Global Optimization*, pages 751–828. Springer US, 1995. doi:10.1007/978-1-4615-2025-2_14.

Anatoliy D. Rikun. A Convex Envelope Formula for Multilinear

Functions. *Journal of Global Optimization*, 10(4):425–437, 1997. doi:10.1023/A:1008217604285.

Ralph T. Rockafellar. *Convex Analysis.* Princeton landmarks in mathematics and physics. Princeton University Press, 2015.

Hermann Schichl and Arnold Neumaier. Interval Analysis on Directed Acyclic Graphs for Global Optimization. *Journal of Global Optimization*, 33(4): 541–562, 2005. doi:10.1007/s10898-005-0937-x.

Martin Schmidt, Denis Aßmann, Robert Burlacu, Jesco Humpola, Imke Joormann, Nikolaos Kanelakis, Thorsten Koch, Djamal Oucherif, Marc E. Pfetsch, Lars Schewe, Robert Schwarz, and Mathias Sirvent. GasLib – A Library of Gas Network Instances. *Data*, 2(4):article 40, 2017. doi:10.3390/data2040040.

Hanif D. Sherali and Amine Alameddine. An explicit characterization of the convex envelope of a bivariate bilinear function over special polytopes. *Annals of Operations Research*, 25(1):197–209, 1990. doi:10.1007/BF02283695.

Hanif D. Sherali and Amine Alameddine. A New Reformulation-Linearization Technique for Bilinear Programming Problems. *Journal of Global Optimization*, 2(4):379–410, 1992. doi:10.1007/BF00122429.

Naum Zuselevich Shor. *Minimization Methods for Non-Differentiable Functions.* Springer Science & Business Media, 2012.

Mohit Tawarmalani. Inclusion Certificates and Simultaneous Convexification of Functions. *Mathematical Programming*, 2010.

Mohit Tawarmalani and Nikolaos V. Sahinidis. Semidefinite Relaxations of Fractional Programs via Novel Convexification Techniques. *Journal of Global Optimization*, 20(2):133–154, 2001. doi:10.1023/A:1011233805045.

Mohit Tawarmalani and Nikolaos V Sahinidis. Convex extensions and envelopes of lower semi-continuous functions. *Mathematical Programming*, 93 (2):247–263, 2002. doi:10.1007/s10107-002-0308-z.

Mohit Tawarmalani and Nikolaos V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.

Mohit Tawarmalani, Jean-Philippe P. Richard, and Chuanhui Xiong. Explicit convex and concave envelopes through polyhedral subdivisions. *Mathematical Programming*, 138(1):531–577, 2013. doi:10.1007/s10107-012-0581-4.

Stefan Vigerske. *Decomposition in Multistage Stochastic Programming and a Constraint Integer Programming Approach to Mixed-Integer Nonlinear Programming*. PhD thesis, Humboldt-Universität zu Berlin, 2012.

Stefan Vigerske and Ambros Gleixner. SCIP: Global Optimization of Mixed-Integer Nonlinear Programs in a Branch-and-Cut Framework. *Optimization Methods and Software*, 33(3):563–593, 2018. doi:10.1080/10556788.2017.1335312.

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