Mixed-Integer Optimization with Ordinary Differential Equations for Gas Networks

Vom Fachbereich Mathematik
der Technischen Universität Darmstadt
zur Erlangung des Grades eines

Doktors der Naturwissenschaften
(Dr. rer. nat.)

genehmigte

Dissertation

von

Oliver Habeck, M. Sc.

aus Neustadt an der Aisch

Referent:	Prof. Dr. Marc E. Pfetsch
Korreferent:	Prof. Dr. Stefan Ulbrich
Tag der Einreichung:	8. Juli 2020
Tag der mündlichen Prüfung:	28. September 2020

Darmstadt 2020

D 17

Acknowledgments

During the past five years in which I have been working on this thesis I got to meet, got to work with, got to know, and even got to become friends with several people. Since all of this would not have been possible if not for the funding of the subproject A01 "Global Methods for Stationary Gas Transport" within the collaborative research center "CRC/Transregio 154 Mathematical Modelling, Simulation and Optimization Using the Example of Gas Networks" (CRC 154) by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG), I gratefully acknowledge the financial support of the DFG.

First and foremost, I would like to thank my supervisor Prof. Marc E. Pfetsch who hired me despite knowing how much programming would be involved and how little programming experience – in particular no knowledge of the programming language C – I had. He always gave me the opportunity to follow my own ideas and spent many hours discussing ideas or problems with me – often about the implementation. Even though we did not always agree on everything, I think that we did work well together and I very much enjoyed working under his supervision.

Furthermore, I want to thank Prof. Stefan Ulbrich. Having finished my Master's thesis under his supervision he told me about project A01 and referred me to Marc Pfetsch. Moreover, his ideas and advice was invaluable for the work in project A01 and for this thesis. In addition, I am also grateful that he accepted to act as a referee for this thesis.

Besides, I would like to thank Prof. Robert Haller-Dintelmann and Prof. Jens Lang for accepting to be part of the examination committee for my thesis.

Many, many thanks go to my current as well as former colleagues in the research group optimization. I always appreciated the (most of the times) good working atmosphere, our joint time in Kleinwalsertal, and the activities beyond work. Several of my colleagues have contributed to this thesis in one or the other way. To begin with, it was a pleasure working with Christopher Hojny. We have not only shared countless tea and coffee breaks, but he also helped me a lot when I started using SCIP. Since then we became friends, had plenty of very helpful discussions, and finally he did proofread a part of my thesis. Furthermore, thanks goes to Frederic

Matter for being such a great office mate and friend, proofreading parts of my thesis, and providing many opportunities for “your mother” jokes. Further, thanks go to Kristina Janzen, Andreas Schmitt, and Anna Walter, especially for proofreading parts of my thesis.

I gratefully look back to travelling with my colleagues from CRC 154. We have had a great time in Berlin, Bamberg, Erlangen, Nürnberg, and especially in Blankensee. In particular, I enjoyed the time with Johann Schmitt who has always been there when I needed someone to talk. Moreover, he ever so often entertained me and all of our colleagues with his stories or shadowboxing.

Finally, I want like to thank Christine Herter. Having just started, when I was almost into my last year of writing this thesis, she is by now one of the most important persons in my life. During the last months she had to put up with me talking a lot about my thesis, and gave me encouragement and moral support, in particular, during some times in which I was quite nervous.

Zusammenfassung

In der vorliegenden Arbeit entwickeln wir einen Spatial Branch-and-Bound Algorithmus zur globalen Lösung einer Klasse von gemischt-ganzzahligen nichtlinearen Optimierungsproblemen, die gewöhnliche Differentialgleichungen als Nebenbedingungen enthalten. Dazu treffen wir die Annahme, dass die Optimierungsprobleme dieser Klasse nur von den Randwerten der Differentialgleichungen abhängen. Dadurch unterscheidet sich diese Klasse und auch unser darauf basierender Ansatz grundlegend von anderen in der Literatur untersuchten Optimierungsproblemen und Methoden. Dabei ist die besondere Struktur der Optimierungsprobleme motiviert durch die Anwendung auf stationären Gastransport. Bei dieser Anwendung lässt sich der Gasfluss auf Rohren durch gewöhnliche Differentialgleichungen beschreiben und es reicht den jeweiligen Gasdruck an den Enden der Rohre zu wissen.

Um die für einen Branch-and-Bound Algorithmus notwendigen Relaxierung der Differentialgleichungen zu konstruieren, untersuchen wir in Kapitel 2 zunächst ein hinreichendes Kriterium, so dass numerische Einschrittverfahren zur Lösung von parameterabhängigen skalaren Anfangswertproblemen untere oder obere Schranken and die exakte Lösung liefern. Daraufhin betrachten wir drei Verfahren und spezifizieren Bedingungen, so dass diese Verfahren untere oder obere Schranken liefern. Zudem untersuchen wir hinreichende Bedingungen unter denen die durch die Verfahren berechneten Approximationen der exakten Lösung konvex oder konkav von den Anfangswerten und Parametern abhängen. Anschließend wenden wir diese Resultate auf die stationären, isothermen Euler-Gleichungen an, welche in unserem Beispiel stationärer Gastransport die Differentialgleichungen sind.

In Kapitel 3 leiten wir dann einen Spatial Branch-and-Bound Algorithmus für unsere allgemeine Klasse von Optimierungsproblemen her. Dazu definieren wir zuerst eine Relaxierung der Differentialgleichungen basierend auf der Annahme, dass wir (mithilfe von numerischen Verfahren wie oben beschrieben) untere und obere Schranken an die exakten Lösungen der Differentialgleichungen berechnen können. Aufgrund der Annahme, dass die Optimierungsprobleme nur von den Randwerten der Differentialgleichungen abhängen, können wir die Relaxierungen so konstruieren, dass diese nur implizit von den Diskretisierungen für die numerischen Einschrittver-

fahren abhängen, d.h. ohne dass wir zusätzliche Variablen für die Diskretisierungen in das Modell aufnehmen müssen. Dies ermöglicht es uns während des Branch-and-Bound Algorithmus die Diskretisierungen adaptiv zu verändern. Dadurch unterscheidet sich unser Verfahren grundlegend von sogenannten *first-discretize-then-optimize* Ansätzen und leidet nicht unter dem Nachteil, dass das Optimierungsproblem für feinere Diskretisierungen immer größer wird.

Wir wenden diesen Algorithmus danach auf das Anwendungsbeispiel stationärer Gastransport an. Dazu präsentieren wir zuerst Modelle für die verschiedenen Elemente eines Gasnetzwerkes. Anschließend zeigen wir, wie wir lineare Relaxierungen für den Gasfluss konstruieren können, und dass die notwendigen Voraussetzungen für eine endliche Terminierung des Spatial Branch-and-Bound Algorithmus erfüllt sind.

Im nächsten Kapitel untersuchen wir verschiedene kombinatorische Modelle mit denen sich die Flussrichtungen von potentialbasierten Flüssen beschreiben lassen. Die Besonderheit von potentialbasierten Flüssen, wie zum Beispiel von stationärem Gas- oder Wasserfluss, ist, dass der Fluss notwendigerweise azyklisch ist, sofern der Druck nicht durch Kompressoren oder Pumpen erhöht wird. Die hauptsächliche Motivation zur Herleitung dieser kombinatorischen Modelle liefern dafür die langen Laufzeiten unseres Algorithmus bei der Anwendung auf stationären Gastransport. Am Ende von Kapitel 5 zeigen wir, dass wir mithilfe der kombinatorischen Modelle die Optimierung von potentialbasierten Energienetzen deutlich beschleunigen können. Dafür verwenden wir wiederum das Beispiel des stationären Gastransports. Allerdings verwenden wir anstatt der Differentialgleichungen ein algebraisches Modell zur Beschreibung des Gasflusses.

In Kapitel 6 präsentieren wir Details unserer Implementierung des Spatial Branch-and-Bound Algorithmus für das Modell mit Differentialgleichungen mit dem Branch-and-Bound Framework SCIP. Danach zeigen wir, dass wir mit unserem Algorithmus, mithilfe der kombinatorischen Modelle und zusätzlichen Techniken zur Verkleinerung von Flussschranken, Optimierungsprobleme auch auf Gasnetzwerken realistischer Größe erfolgreich und effizient lösen können.

Abstract

This thesis deals with the development of a spatial branch-and-bound algorithm for global optimization of a class of mixed integer nonlinear problems including ordinary differential equation (ODE) constraints on an underlying network structure. The distinguishing feature of this class is that the ODE solutions only need to be known at a finite number of points, that is, the junctions of the underlying network. Instead of using a first-discretize-then-optimize approach, we show that we can compute lower and upper bounds on the solutions of initial value problems by using appropriate discretization methods. To construct relaxations of ODE constraints, we derive sufficient conditions under which a discretization method yields a lower or an upper bound and apply the result to specify the conditions for particular methods. Exploiting the underlying network structure, we use these methods to define under- and overestimators of the ODE solutions. Moreover, we derive conditions that ensure the convexity or concavity of the obtained under- and overestimators. The underlying network structure, enables us to incorporate the relaxations defined by discretization methods into the mixed-integer optimization problem without introducing new variables for the discretization. This in turn makes it possible to use the relaxations in a spatial branch-and-bound process which allows to adaptively refine the discretizations. With this algorithm we can compute global ε-optimal solutions or decide infeasibility for optimization problems of the class above. Furthermore, we prove that this algorithm terminates finitely under some natural assumptions on the under- and overestimators.

Then we apply our spatial branch-and-bound algorithm to the example of stationary gas transport to show that the approach works. To speed-up the optimization process we introduce problem specific bound tightening methods based on the discretizations and moreover we investigate acyclic flows. To this end, we consider potential-based flows which are a basic model to represent energy networks. In passive networks potential-based flows are necessarily acyclic. Based on binary variables for flow directions, we introduce several combinatorial models for acyclic flows. We study in particular one model that captures acyclicity together with the supply and demand behavior of the network. We analyze properties of this model, including

variable fixing rules and the complexity of linear optimization over the corresponding polytope. Using this model we can solve optimization problems on a gas network including almost 300 ODE constraints with a geometric mean time of about nine minutes to prove optimality.

Contents

CHAPTER 1

Introduction

This thesis deals with the development of a new method for global optimization of mixed-integer nonlinear problems including ordinary differential equations (ODEs). The source of both motivation and inspiration for the particular structure of the investigated problems as well as the methods and algorithmics is the application on stationary gas transport. This is the ever-present application in this thesis, since the thesis originated from the subproject A01 “Global Methods for Stationary Gas Transport” within the collaborative research center “CRC/Transregio 154 Mathematical Modelling, Simulation and Optimization Using the Example of Gas Networks” (CRC 154) funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG).

In 2014 the DFG decided to fund the CRC 154 which was proposed by researchers from the universities Friedrich-Alexander-Universität Erlangen-Nürnberg, Humboldt-Universität zu Berlin, Technische Universität Darmstadt, and Universität Duisburg-Essen and the research institutes Weierstraß-Institut für Angewandte Analysis und Stochastik Berlin, and Zuse Institute Berlin. Moreover, in 2018 the DFG extended the funding for additional four years. In the broader sense, these decisions have been made since (energy efficient) gas transport became of particular interest to the general public and is a relevant application for different parts of (applied) mathematics.

In 2011 the nuclear disaster in Fukushima triggered the energy turnaround in Germany. As a part of that eight nuclear power plants were decommissioned and the total nuclear power phase-out by the end of the year 2022 was decided. In the transition period, until enough renewable and sustainable energy can be produced to completely replace nuclear and fossil fuel driven power production, gas is considered to play an important role. In the medium term there are sufficient gas resources, the transport is quite safe, and gas cannot only be used to generate power but for heating, too. Furthermore, since gas driven power plants are able to start up and shut down quickly, they are better suited to react on power and demand fluctuations in the energy network to ensure a stable and sufficient energy supply than, for example, coal-fired power plants. Such power fluctuations may for example be caused by rapid weather changes which directly impact the production of renewable energy through wind or solar power plants. Moreover, the gas network itself can be used as an energy storage by inserting additional gas. To this end, not only natural gas can be injected but also green hydrogen which can be produced if there is a surplus of renewable energy. The importance of green hydrogen has been emphasized by the recently made public *national hydrogen strategy* of the German government [14]. Furthermore, in the long run gas networks can eventually be repurposed to transport hydrogen instead of natural gas; see [150]. Note that the production of green hydrogen is not a scenario far-off in the future. The Dutch gas transport company Gasunie [42], which operates a gas network in the Netherlands and northern Germany, has opened the first one megawatt green hydrogen plant called “HyStock” near Groningen in June 2019. Hence, gas *transmission system operators* face new challenges, all the while safe, reliable and energy efficient gas transport is desired. However, the daily operation and long-term planning of gas networks is still very strong oriented on experience and simulation tools.

With the different levels of detail of physical models for the gas flow stationary as well as instationary gas transport is an interesting application from a mathematical point of view. Gas transport posed many challenges and open questions for different parts and communities of (applied) mathematics back in 2014 and still does in 2020; for example, see Hante et al. [58]. Within the CRC 154 different questions have been and are addressed. These questions range from the development of new techniques for robust or stochastic optimization with algebraic models, over the existence of solutions for and the global optimization with ordinary differential equations, to finding appropriate coupling or optimality conditions and developing large-scale simulation methods for high detailed models involving hyperbolic partial differential equations. Moreover, the developed models and methods are typically not only applicable to the specific application of gas transport but to a broader class of problems and energy networks. Furthermore, also the modeling and analysis of gas markets with detailed models of the gas flow is of particular interest in the field of economics.

Due to its public and mathematical relevance the DFG decided to accept the research proposal for the CRC 154 including the subproject A01. With its goal to develop methods for global optimization of stationary gas transport the research proposal of subproject A01 is the starting point of this thesis. Since the application on stationary gas transport will be used throughout this thesis, we start by introducing the particular ordinary differential equation, which we use to describe the gas flow in a pipeline, and the modeling choices under which this differential equation is derived.

1.1 The Physics of Gas Flow in Pipelines

In the context of gas transport optimization or simulation, gas flow through pipelines is usually described by the one-dimensional Euler equations or a model derived from them through simplifications. The Euler equations themselves are a system of nonlinear hyperbolic partial differential equations (PDEs), which are derived from the Navier-Stokes equations; see Feistauer [36]. The Euler equations for a single pipeline consist of the *continuity equation*, the *momentum equation*, and the *energy equation*

$$\begin{aligned} \partial_t \rho + \partial_x(\rho v) &= 0, \\ \partial_t(\rho v) + \partial_x(p + \rho v^2) &= -\frac{\lambda}{2D}\rho v|v| - g\rho\sigma, \\ \partial_t\left(\rho(\tfrac{1}{2}v^2 + e)\right) + \partial_x\left(\rho v(\tfrac{1}{2}v^2 + e) + pv\right) &= -\frac{k_w}{D}\left(T - T_w\right), \end{aligned} \tag{1.1}$$

together with appropriate initial and boundary data. In these constraints the unknowns are the *density* $\rho(x,t) \in \mathbb{R}_{\geq 0}$ in $\mathrm{kg/m^3}$, the *velocity* $v(x,t) \in \mathbb{R}$ in direction of the pipeline in m/s, the *pressure* $p(x,t) \in \mathbb{R}_{>0}$ in Pa, and the *temperature* $T(x,t) \in \mathbb{R}_{>0}$ of the gas in K. Moreover, $e(x,t) = c_V T(x,t) + g\,h(x)$ denotes the *internal energy* as the sum of thermal and potential energy. The parameters are the *friction coefficient* λ, the *diameter* D of the pipeline in m, the *slope* $\sigma \in [-1,1]$ of the pipeline, the *height* $h(x)$ of the pipeline in m, the *specific heat capacity* c_V of the gas in J/(kg K), the *heat transfer coefficient* k_w between the gas and the pipelines wall in $\mathrm{J/(s\,m^2\,K)}$, and the *temperature of the pipelines wall* $T_w(x,t) \in \mathbb{R}_{>0}$ in K. Furthermore, g denotes the *gravitational acceleration* in $\mathrm{m/s^2}$, e.g., the standard gravitational acceleration of $9.806\,65\,\mathrm{m/s^2}$.

The Euler equations with its four unknowns and three equations are completed by the *thermodynamical standard equation for real gases*

$$p = \rho\, R_s\, T\, z \tag{1.2}$$

with the *specific gas constant* R_s in J/(kg K) and the *compressibility factor* z of the gas.

Before we derive our model from the equations (1.1) and (1.2), note that these already contain some modeling choices, which we will keep throughout this thesis. First of all, they are formulated for a single composition of gas and not for a mixture of several gas types, i.e., we assume that the gas in the network and the inflow at the entries does not vary in *calorific value*, *molar mass*, *norm density*, and so forth. We assume that the slope σ is constant along a pipeline. Furthermore, there are several approximations and (empirical) formulas for the friction coefficient and the compressibility factor known in the literature. We use the formula of Nikuradse [106, 107]

$$\lambda = \left(2\log_{10}\left(\frac{D}{\kappa}\right) + 1.138\right)^{-2} \tag{1.3}$$

to determine the friction coefficient. Here, κ denotes the *integral roughness* in m of the pipelines inner surface. Later, in Section 4.5 we will discuss how to incorporate flow dependent formulas for the friction coefficient into our algorithmic framework. To compute the compressibility factor, we use the formula of the American Gas Association (AGA)

$$z(p,T) = 1 + 0.257\frac{p}{p_c} - 0.533\frac{p}{p_c}\frac{T_c}{T} \tag{1.4}$$

see Králik et al. [84]. Here, p_c and T_c denote the *pseudocritical pressure* and *pseudocritical temperature* of the gas.

In order to derive the model we will use throughout this thesis, we make the following two major assumptions.

1. We consider *stationary gas transport*, i.e., changes over time are neglected.
2. We consider the *isothermal* case, i.e., we assume the gas has a *constant mean temperature* T_m.

We use these assumptions for the following reasons. The planning of gas transport and the operation of a gas network includes various time scales, which is also reflected in the contracts for gas transportation. In Germany, there are *long-term* contracts over weeks or even years, but also *short-term* contracts where customers can buy the right to inject or withdraw gas the day before it actually happens. Furthermore, the transmission system operators have to react on fluctuations in consumption or supply during the day; for an overview on gas markets and rules in Germany see Chapter 3 in Koch et al. [82]. In long-term planning, it is a common assumption to consider stationary gas transport, for instance, since the exact state of the network, demands, supply, and weather conditions are not known. The second assumption of constant gas temperature is also very common, especially in long-term planning, when only statistical data on temperature is available. Furthermore, in Germany gas pipelines are often under ground, which additionally reduces temperature fluctuations of the surroundings.

For (1.1) these two assumptions imply dropping the time derivatives and the energy equation. Then the equations read

$$
\begin{aligned}
\partial_x(\rho v) &= 0, \\
\partial_x(p + \rho v^2) &= -\frac{\lambda}{2D}\rho v|v| - g\rho\sigma.
\end{aligned}
\tag{1.5}
$$

To rewrite these equations in terms of pressure p and *mass flow* q in kg/s instead of p, ρ, and v, we use the formula $q = A\rho v$ for mass flow in cylindrical pipelines with cross-sectional area A and assume a constant compressibility factor z_m. To this end we define $z_m := z(p_m, T_m)$ using a mean pressure value p_m. With the lower and upper pressure bounds $\underline{p}_l$, $\overline{p}_l$ and $\underline{p}_r$, $\overline{p}_r$ on the left and right boundary of the pipeline, which are given in our test data (see Section 6.3), we use the formula

$$
p_m := \frac{1}{2}\max\{\underline{p}_l, \underline{p}_r\} + \frac{1}{2}\min\{\overline{p}_l, \overline{p}_r\}
\tag{1.6}
$$

for p_m as suggested by Geißler et al. [45].

As a consequence of the constant compressibility factor, the *speed of sound* c in m/s in the gas is also constant. In general, the speed of sound is given by $c = \sqrt{\partial_\rho p}$, but by using the equation of state (1.2) with constant compressibility factor z_m, we deduce that $c = \sqrt{R_s T_m z_m} = \sqrt{\rho^{-1} p}$. With that, we can express (1.5) in terms of p and q by

$$
\begin{aligned}
\partial_x q &= 0, \\
\partial_x\left(p + \frac{c^2 q^2}{A^2 p}\right) &= -\frac{\lambda c^2}{2DA^2}\,\frac{q|q|}{p} - \frac{g}{c^2}\sigma\, p.
\end{aligned}
\tag{1.7}
$$

The continuity equation reduces to the mass flow q being constant and thus in the stationary isothermal setting the former hyperbolic PDE system turns into a scalar ordinary differential equation of the form

$$
\partial_x p\left(1 - \frac{c^2 q^2}{A^2 p^2}\right) = -\frac{\lambda\, c^2}{2DA^2}\,\frac{q|q|}{p} - \frac{g}{c^2}\sigma\, p.
\tag{1.8}
$$

Note that the fraction $\frac{c^2 q^2}{A^2 p^2}$ is equal to $\left(\frac{v}{c}\right)^2$. Since the gas typically travels with a velocity which is much smaller compared to the speed of sound, this term is often neglected; see Osiadacz [108]. In fact, neglecting this term is used to derive the Weymouth equation, which is, according to Ríos-Mercado and Borraz-Sánchez [114], "most-widely used to model flow capacities." For $\sigma = 0$ the Weymouth equation is given by

$$p(0)^2 - p(L)^2 = \left(\frac{4}{\pi}\right)^2 \frac{L}{D^5}\, \lambda\, c^2\, q|q|, \tag{1.9}$$

where L is the length of the pipe in m and $p(0)$, $p(L)$ are the pressure values at both ends of the pipe. For a derivation of this equation see, for example, Koch et al. [82]. We will use this algebraic model for the computational results in Chapter 5, but the most parts of this thesis deal with global optimization of ODE constraint optimization problems. Therefore, we consider a more detailed ODE model instead and assume that $\frac{|v|}{c} \leq \nu_c$ holds for a constant $\nu_c \in (0, 1)$. Then solving equation (1.8) for $\partial_x p$ yields

$$\partial_x p = -\frac{1}{2} \frac{\lambda\, c^2\, q|q|\, p}{D(A^2 p^2 - c^2 q^2)} - \frac{g \sigma A^2 p^3}{c^2 (A^2 p^2 - c^2 q^2)}. \tag{1.10}$$

Throughout this thesis we will denote the right-hand side with φ_σ. Finally, the ODE we use to describe the gas flow through a pipeline is

$$\partial_x p(x) = \varphi_\sigma\big(p(x), q\big) \coloneqq -\frac{1}{2} \frac{p(x)}{c^2 D} \frac{\lambda c^4 q|q| + 2 D g \sigma A^2 p(x)^2}{A^2 p(x)^2 - c^2 q^2}, \quad x \in [0, L]. \tag{1.11}$$

Since we will often consider the case without height differences (first), i.e., $\sigma = 0$, we use the shorthand notation $\varphi = \varphi_0$.

1.2 Goals and Fundamental Ideas

The main goal formulated in the proposal of project A01 and also of this thesis is the development of a global optimization method for problems constrained by ODEs such as the stationary isothermal Euler equation (1.8). A very natural approach for solving ODE or PDE constrained optimization problems is to discretize the differential equations and then solve the resulting mixed-integer nonlinear program (MINLP) with a fixed discretization to global optimality, e.g., by spatial branch-and-bound which is a standard method for global optimization. However, this only yields solutions with an a priori fixed accuracy and this does not define a relaxation of the original problem, i.e., exact solutions of the ODEs or PDEs typically are infeasible for the discretized MINLP. To overcome the former problem, one can perform an a posteriori accuracy check and, if it fails, refine the discretization and solve the adjusted MINLP again. A problem with this approach is that the MINLPs become very large for fine discretizations. Moreover, in general we cannot expect convergence of the solutions even if the discretization step sizes tend to 0; Hante and Schmidt [60] provide sufficient conditions for the convergence of the optimal value of ODE constrained problems.

Our goal is to develop a method which combines adaptive discretization with spatial branch-and-bound in a single algorithm, i.e., without repeatedly solving MINLPs by spatial branch-and-bound. To define relaxations of the ODE constraints which are necessary for spatial branch-and-bound the following idea has been sketched in the original proposal of subproject A01. Discretizing the differential equations, here (1.7), results in a nonlinear equality system, for example,

$$\frac{p_i - p_{i-1}}{x_i - x_{i-1}} + \frac{c^2\, q^2}{A^2}\left(\frac{p_i^{-1} - p_{i-1}^{-1}}{x_i - x_{i-1}}\right) + \frac{\lambda c^2}{2DA^2} q|q| \frac{1}{p_i} + \frac{g}{c^2}\sigma\, p_i = 0 \qquad \forall i \in [N],$$

where we have $0 = x_0 < x_1 < \cdots < x_N = L$ and $[N] := \{1, \ldots, N\}$ for $N \in \mathbb{N}$. Then, if we can derive appropriate error estimates for the error produced by discretization, we can turn this equality system into an inequality system by adding and subtracting the error estimator. Thus, for a fixed discretization this yields a relaxation of the original problem which can be solved by state-of-the-art MINLP solver. However, to control the quality of the relaxation, the discretization has to be refined.

Consider a general one-step method for solving initial value problems, i.e., a discretization method which can be formulated as

$$p_0 = p(0), \quad p_i = p_{i-1} + h_i\, \varphi_h\big(x_{i-1}, h_i, q, p_{i-1}, p_i\big) \quad \forall i \in [N] \tag{1.12}$$

with *increment function* φ_h and $h_i = x_i - x_{i-1}$ for $i \in [N]$. A naive idea is to use the *local discretization error* as error estimator for such methods. The local discretization error for a one-step method such as (1.12) is

$$\tau(x, h) = p(x + h) - p(x) - h\, \varphi_h\big(x, h, q, p(x), p(x + h)\big),$$

i.e., the error produced by performing one discretization step. Then, if we can derive a bound on the local discretization error, $|\tau(x, h)| \le \xi$, as tight as possible, we can simply add $\pm\xi$ to the iteration rule of the one-step method, that is, we get the inequality system

$$p_{i-1} + h_i\, \varphi_h\big(x_{i-1}, h_i, q, p_{i-1}\big) - \xi \ \le\ p_i \ \le\ p_{i-1} + h_i\, \varphi_h\big(x_{i-1}, h_i, q, p_{i-1}\big) + \xi$$

for $i \in [N]$ and the the exact solution of the ODE would define a feasible solution through $p_i = p(x_i)$ for $i \in [N]$. But the problem with this is that due to adding $\pm\xi$ for every step the feasible set for p_i grows funnel-like in i. Since the local discretization error usually decreases for step sizes $h_i \to 0$, we can control the error of the relaxation by refining the discretization. However, this requires that we introduce additional variables in the optimization model. To the best of our knowledge it is still an open question whether and how this can be done in a single spatial branch-and-bound algorithm, i.e., without having the restart spatial branch-and-bound.

Our fundamental ideas on how to overcome these problems are based on the observation that our optimization problem (which we will introduce in Chapter 4) only depends on the boundary values of the ODE (1.11). That is, for every pipeline we have constraints of the form

$$\begin{aligned} &\partial_x p(x) = \varphi_\sigma\big(p(x), q\big), \quad x \in [0, L], \\ &p_u = p(0),\ p_v = p(L), \end{aligned} \tag{1.13}$$

where p_u and p_v are variables which represent the pressure levels at both ends of the pipelines. Other than that our model is an MINLP, which does not depend on the solution of the differential equations. Thus, if we would have an algebraic formula for the analytical solution of (1.11), then we could replace the ODE constraints by the solution operator $P(p_u, q) = p_v$ which maps initial value $p(0) = p_u$ and mass flow q to the solution at the boundary $p(L) = p_v$. Note that there actually exist analytical solutions of (1.11), however, evaluating them requires numerical evaluation for example by Newton's method and hence they are not suited for the use in standard spatial branch-and-bound approaches; see Gugat et al. [54]. Nevertheless, if we can define an appropriate relaxation $\mathcal{R}$ of the feasible set

$$\mathcal{F} := \Big\{(p_u, p_v, q) \in \big[\underline{p}_u, \overline{p}_u\big] \times \big[\underline{p}_v, \overline{p}_v\big] \times \big[\underline{q}, \overline{q}\big] \ : \ (p_u, p_v, q) \text{ satisfy (1.13)}\Big\}$$

defined by the ODE constraints (1.13), then we could still replace the ODE constraints by the relaxation $\mathcal{R}$.

Our main idea to define a suitable relaxation $\mathcal{R}$ is the following. Instead of using the local discretization error to relax the discretization, suppose that we can prove that a numerical one-step method (1.12) actually produces a lower or an upper bound on the analytical solution, that is, we are able to prove that either the inequality $p_N \leq p(L)$ or $p_N \geq p(L)$ holds. In the case that we can find two methods such that one produces a lower bound and one produces a upper bound, we can use this in a spatial branch-and-bound approach as follows. In every node of the branch-and-bound tree, we solve a convex or linear relaxation of the original problem. If this relaxation is feasible, then we check for each pipeline if the corresponding solution is close to the exact solution. Therefor, we evaluate the two methods with the mass flow and one pressure value in the solution. If the difference of the lower and upper bound satisfies a given tolerance and the other pressure value is within the range of the lower and upper bound, we say that the solution is approximatively feasible (δ-feasible). If the difference does not satisfy the a priori tolerance and both methods are convergent, we refine the discretization and recompute the lower and upper bounds until the tolerance holds. Moreover, if the solution is not approximatively feasible, the hope is that we can cut the solution off based on information given by the evaluation of the two one-step methods.

Note that in this idea the discretization is only implicitly used to check feasibility of solutions and to separate infeasible solutions, that is, the optimization problem does not include variables to explicitly represent the discretization. Nevertheless, finer discretizations still yield tighter bounds. Hence, this enables us to adaptively change the discretization to compute solutions with the desired accuracy.

1.3 Outline of the Thesis

The structure of this thesis partially follows the fundamental ideas presented in the previous section. In Chapter 2 we investigate numerical methods which produce lower or upper bounds for the solution of initial value problems. To this end, we first provide sufficient conditions under which one-step methods produce lower and upper bounds for scalar ODEs. Then we apply this result to three particular one-step methods and specify the conditions such that these particular methods produce lower or upper bounds. Moreover, we study the properties of the corresponding input-output functions, i.e., the functions which map the initial values and parameter to the approximation at the second boundary. For the example above, the input-output function is the function $(p_0, q) \mapsto p_N$. Then in Sections 2.3 and 2.4, we apply the results to stationary gas transport, that is, the ODE (1.11). First we consider only horizontal pipelines, i.e., the case $\sigma = 0$, and afterwards extend the results to the general case. In Section 2.5, we present a result which extends the sufficient conditions from scalar ODEs to ODE systems and discuss an idea how to combine different one-step methods to compute lower and upper bounds if these methods only produce bounds for specific parts of the ODE solution.

In Chapter 3, we introduce a particular class of mixed-integer nonlinear optimization problems containing ODE constraints motivated by the structure of the stationary gas transport problem and develop a general spatial branch-and-bound framework to compute δ-feasible solutions for such problems. Therefor, we first reformulate and define a relaxation of this problem class in Section 3.2 based on the assumption that under- and overestimators for the analytical solution of the differential equations exists. Then we show how δ-feasible solutions of the relaxation and the reformulation and thus the original problem as well are related. This gives rise to two approaches similar to first-discretize-then-optimize, i.e., either use a fine discretization and solve the resulting problem once or iteratively solve an MINLP, check differences of lower and upper bounds in the solution, and refine the discretization if necessary. We show that both approaches work under mild assumptions in Section 3.3. However, we show that the idea presented above to combine the feasibility check and refining the discretization in a single spatial branch-and-bound tree yields a finitely terminating spatial branch-and-bound algorithm in Section 3.4.

We apply this framework to the example of stationary gas transport in Chapter 4. There we introduce the models for different network elements first. Based on the results from Section 2.3, we show how we can derive a linear relaxation of the gas flow on pipelines without height differences via the explicit midpoint method and the implicit trapezoidal rule in Section 4.3. Then we prove that this particular relaxation satisfies the necessary requirements such that the spatial branch-and-bound framework terminates finitely. Afterwards, in Section 4.5, we show how to extend this approach such that we can also use models with a nonconstant friction coefficient (see (1.3)) and handle pipelines with height differences. Finally in Section 4.6, we present first numerical results for global optimization on a small gas network.

Motivated by the numerical results and the observation that in our stationary setting gas cannot flow in cycles (unless pressure is increased in compressor stations), we develop and investigate combinatorial models for acyclic flows in Chapter 5. Since not only stationary gas flow is acyclic but more generally so-called potential-based flows are acyclic, we investigate potential-based flows there. To this end, we introduce binary variables which are coupled with the flow variables and represent the flow directions. Moreover, we introduce a nested sequence of polytopes which provide combinatorial models for the flow directions in Section 5.3. Thereby, these polytopes relax more and more of the nonlinear constraints of potential-based flows. Subsequently, we study their relation and investigate the complexity of optimizing over a polytope which is solely based on the flow direction variables. In Section 5.3.3, we then introduce our main combinatorial model, exploiting both acyclicity and the fact that one needs to connect sources and sinks. To see that this model provides a good compromise between the nonlinear model and the so-called acyclic subgraph polytope, the particular model is investigated in Sections 5.3.4 and 5.3.5 in more detail. Then in Section 5.4, we use this model for optimization of a stationary gas transport problem with the potential-based flow model instead of the ODE model. We demonstrate that this approach leads to an improvement of the total geometric mean solving time by about a factor of 3, a factor of about 5 in the geometric mean time to prove optimality, and a significant speed-up for the total computational time with a factor of 7.

In Chapter 6 we present the implementation and numerical results of our spatial branch-and-bound algorithm for the ODE constrained stationary gas transport model. We discuss problem specific bound tightening methods based on the lower and upper bounds derived by the explicit midpoint method and trapezoidal rule, and a variant of optimality based bound tightening in Section 6.2. Then in Section 6.3

we present the computational setting and test data. We discuss some numerical problems we are facing and investigate the influence of the objective function and compressor station model on the computational performance and the results. In the end of Chapter 6 we present a comprehensive computational study of the effect of our bound tightening techniques and the combinatorial model for handling acyclic flows. We show that these significantly improve the performance of our algorithm such that we can efficiently solve optimization problems on a gas network of realistic size.

Finally, we end this thesis with a conclusion and an outlook in Chapter 7.

1.4 Literature Review

In the following, we summarize some literature related to our goal of developing a spatial branch-and-bound algorithm for solving ODE constrained optimization problems, the application of stationary gas transport and potential-based flows. We remark that Chapters 2 to 5 contain more detailed literature reviews specifically dealing with the topics addressed in these chapters.

Spatial branch-and-bound is a standard method for global optimization of (nonconvex) MINLPs and is typically used by state-of-the-art solvers for such problems; for example, see Horst and Tuy [73], and Kılınç and Sahinidis [79]. Moreover, spatial branch-and-bound is often used within first-discretize-then-optimize approaches for ODE or PDE constrained problems, that is, after discretizing the original problem spatial branch-and-bound is applied to solve the resulting MINLP to global optimality; see, e.g., Gerdts [46]. However, the solutions only provide an approximation of the exact solutions with respect to an a priori fixed accuracy. For the case of ODE constrained problems including integer variables Hante and Schmidt [60] provide sufficient conditions such that at least the optimal values converge to the exact optimal value, if the discretizations are iteratively refined and the resulting MINLPs solved again. The drawback of first-discretize-then-optimize approaches is that the MINLPs become very large for high precision and thus harder to solve.

During spatial branch-and-bound convex or linear relaxations of the original problem are solved on increasingly smaller parts of the feasible set. To this end, there exist various convexification techniques; for example, see McCormick [99], Adjiman and coworkers [3, 4], or Sherali and coworkers [137, 138, 139]. To derive relaxations of ordinary differential equations (over time) there exist two major approaches called *time-discretization techniques* and *continuous-time enclosure techniques*. The basic idea of the first approach is based on discretizing the ODEs and then relaxing the result; see, for example, Nedialkov et al. [104], Neher et al. [105], or Sahlodin and Chachuat [124, 125]. The main idea of continuous-time enclosure techniques is to construct an auxiliary ODE system such that its solutions provide lower and upper bounds; for example, see Singer and Barton [140], Scott and Barton [134], or Harwood and Barton [63]. Moreover, these methods can be used for global optimization of dynamic systems, e.g., see Chachuat et al. [21, 22] or Lin and Stadtherr [91], however, they do not fit in with our ideas presented in Section 1.2. In the time-discretization approaches the solved problems explicitly depend on the discretization and in the continuous-time enclosure approaches the auxiliary ODE systems still have to be solved with arbitrary precision. Furthermore, Bajaj and Hasan [6] present a deterministic global optimization algorithm for dynamical systems by constructing edge-concave underestimators for functions depending on ODE solutions. Thereby, Bajaj and Hasan extend the work by Hasan [64].

The problem class which we consider in Chapter 3 is related to mixed-integer optimal control problems with ODEs as well as PDEs. A starting point for literature on optimal control with ODEs see the following list. Sager et al. [120] developed a convexification method to handle discrete decisions over time that switch the right-hand sides of ODEs and show how to efficiently compute feasible solutions. This method has been further investigated and extended in a series of articles by Sager and coworkers [119, 121], Jung et al. [76], Zeile and coworkers [122, 156, 157], and Kirches et al. [80].

For a starting point on literature on optimal control including PDE constraints see, for example, Hinze et al. [70] and the following articles. Buchheim et al. [17] present a global approach for solving particular semilinear elliptic mixed-integer PDE problems with distributed and boundary control using outer-approximation. Moreover, the convexification method by Sager et al. [120] has been extended to PDE constrained problems by Hante and Sager [59], Göttlich et al. [48], Hahn et al. [57], and Manns and Kirches [95].

Throughout this thesis, we will use the application on stationary gas transport as example. We refer to Koch et al. [82] and Ríos-Mercado and Borraz-Sánchez [114] for general information on modeling and solution methods for gas transport problems. Moreover, Hante et al. [58] present challenges and open problems for gas and fluid flow in networks. Furthermore, the articles by Gugat et al. [53] and Schmidt et al. [128] propose two approaches for solving ODE constraint optimization problems, too. Since they are applied to a stationary gas transport problem similar to ours they will be discussed in more detail in Chapter 4.

In Chapter 5, we investigate combinatorial models for acyclic flows which can be used to speed-up the optimization of potential-based flows. There, we exploit that potential-based flows are necessarily acyclic. For an overview on potential-based flows see Hendrickson and Janson [67]. Furthermore, an important existence and uniqueness result for potential-based flows can be found in the articles by Maugis [98], Collins et al. [26], and Ríos-Mercado et al. [115]. Acyclic flows are also studied by Becker and Hiller in the articles [7, 8, 9, 68], however, by a different approach than ours. We discuss the differences in Section 5.1.

We remark that Chapters 2 and 3 and parts of Chapter 4 are based on the results presented in the article [56] which is joint work with Marc E. Pfetsch and Stefan Ulbrich. Moreover, Chapter 5 has been published online in similar form in the article [55] which is joint work with Marc E. Pfetsch, too. Furthermore, this article is submitted for publication in an international journal.

1.5 Scientific Contribution

The main scientific contribution of this thesis is the development of a spatial branch-and-bound algorithm for global optimization of a particular class of ODE constrained optimization problems in Chapter 3. Due to the assumption of an underlying network structure, i.e., the ODEs have to hold on arcs and are coupled to the optimization problem only through parameters and boundary values, this class of optimization problems is distinct from other approaches for mixed integer optimal control which often deal with integer decisions over time, for example, optimal gear shifting. Our algorithm is based on standard methods from numerical analysis and integer as well as nonlinear optimization, that is, one-step methods for solving initial value problems and spatial branch-and-bound for global optimization of mixed-integer nonlinear problems, however, they are combined in a novel way. The main mathematical challenge is the construction of relaxations for ODE constraints and their integration into spatial branch-and-bound. To this end, we define relaxations of ODE constraints implicitly based on one-step methods for which we prove in Chapter 2 that they produce lower and upper bounds on the solutions of initial value problems and that their corresponding input-output functions are convex or concave. The particular construction of the relaxations enables us to use adaptive discretizations of ODEs within a single spatial branch-and-bound tree. To the best of our knowledge neither have these specific properties of the input-output functions been investigated before nor exists another algorithm which combines spatial branch-and-bound and adaptive discretization of ODEs.

In Chapter 5, we develop and investigate combinatorial models for acyclic flows. Therefore, we study different polytopes which provide such combinatorial models and investigate the complexity of optimizing over some of them. We show that the models can be used to significantly speed-up global optimization of potential-based flows. Our computational results for the example of stationary gas transport (with the potential-based flow model) show a speed-up factor of about 5 in the geometric mean time to prove optimality and even a speed-up factor of about 7 in the total running time.

Combining our spatial branch-and-bound algorithm for ODE constrained problems with the combinatorial models for acyclic flows and problem specific bound tightening techniques yields a fast solver for global optimization of stationary gas transport problems with a more detailed model than usually considered in the literature; see Chapters 4 and 6. We have put a lot of effort into the implementation of the algorithm and the additional techniques with the branch-and-bound framework SCIP. As a consequence, we can solve optimization problems on a gas network with 582 nodes and almost 300 pipelines, i.e., ODE constraints, with a geometric mean time of about 9 minutes to prove optimality and less than 100 seconds to prove infeasibility. The same network has been used to present other global optimization approaches on the example of stationary gas transport, too, however, they use an algebraic potential-based flow model instead of the more complex ODE model we use; for example, see Pfetsch et al. [111], Koch et al. [82], or Burlacu et al. [20].

CHAPTER 2

Bounding the Solutions of Ordinary Differential Equations

In the introduction, we presented an idea how to use numerical one-step methods to define relaxations for ordinary differential equation constraints of optimization problems. To concretize this idea consider constraints

$$\begin{aligned}
&\partial_s y(s) = f(s, x, y(s)), \quad s \in [0, S], \\
&y(0) = y^0,\ y(S) = y^S, \\
&x \in X,\ y^0 \in Y^0,\ y^S \in Y^S,
\end{aligned} \tag{2.1}$$

given by a parameter-dependent differential equation with continuously differentiable right-hand side f, parameters x and solution $y(s)$, which is coupled with variables y^0 and y^S. Let $X \subset \mathbb{R}^k$ and Y^0, $Y^S \subset \mathbb{R}^n$ be bounded and convex. Our working assumption is that the differential equation is only coupled to the remaining constraints of the optimization problem by these constraints, i.e., the rest of the problem only depends on the variables x, y^0 and y^S, but not on the solution $y(s)$ for $s \in (0, S)$.

Suppose that the ODE admits an analytical solution and let $F\colon X \times Y^0 \to Y^S$ be an algebraic function which describes the relation between parameters x, initial values $y(0)$ and end values $y(S)$, i.e., $F(x, y(0)) = y(S)$. Then, for example, if F can be decomposed in an expression tree (see, e.g., Smith and Pantelides [144]), we can replace the ODE constraint above by $F(x, y^0) = y^S$ and solve the resulting MINLP by standard techniques.

Our idea comes into play, if no such function F exists or it cannot be treated by standard techniques; e.g., see Gugat et al. [54] who show that an analytical solution of (1.11) exists but it has to be evaluated numerically, for example, by Newton's method. If we can find two one-step methods which provably yield a lower, respectively, an upper bound on the exact solution $y(s)$, then the corresponding input-output functions F^ℓ and F^u, i.e., the functions which map parameters and initial value to the computed approximation of $y(S)$, satisfy $F^\ell \leq F \leq F^u$. Hence, we could use these functions to define a relaxation of the ODE constraint via

$$F^\ell(x, y^0, N) \leq y^S \leq F^u(x, y^0, N),$$

where N denotes the number of grid points in the discretization,

To investigate this idea the chapter is structured as follows. We start with a literature review of existing approaches to enclose the solution of ODEs. Then, in Section 2.2 we study one-step methods for scalar initial value problems. After providing a sufficient criterion such that one-step methods produce lower or upper bounds, we apply this result to three particular one-step methods. Moreover, we characterize conditions under which the corresponding input-output functions are convex or concave. We show that these methods can be used to compute lower and upper bounds on the gas flow in horizontal pipes in Section 2.3 and extend the results to nonzero slope in Section 2.4. Finally, in Section 2.5 we discuss how to perform bound propagation if the properties of the ODE solution are not favorable and discuss the extension to ODE systems.

We point out that the results presented in this chapter are a revised and extended version of results that were already published in [56] which is joint work with Marc E. Pfetsch and Stefan Ulbrich. Section 2.2 is extended by further remarks, in particular, Remark 2.8, which discusses the convergence of the considered methods. Note that the convergence of the particular methods will be important for the spatial branch-and-bound algorithm which we develop in Chapter 3. Furthermore, the results in Sections 2.3 and 2.4 have been generalized.

2.1 Literature Review

Generating tight enclosures for solutions of ordinary differential equations is of great interest for global optimization of dynamic systems; see, for example, Chachuat et al. [21, 22] or Lin and Stadtherr [91]. In the literature there are two major approaches to compute such enclosures for parametric initial value problems, namely time-discretization techniques and continuous-time enclosure techniques.

The basic idea of time-discretization techniques is similar, though more sophisticated, to the idea of using the local discretization error to derive bounds on the ODE solution, which was sketched in Section 1.2. After discretizing the differential equation, these methods (try to) compute tight bounds on the approximation error. Several publications about this topic are based on the work of Moore [103], who used interval analysis to test if a solution of an ODE exists and is unique over a finite time step. Nedialkov et al. [104] review such methods using interval arithmetic. Other methods based on Taylor models (often) in combination with interval arithmetic for parametric ODEs have been developed by Neher et al. [105], Lin and coworkers [90, 92], Sahlodin and Chachuat [124, 125], and Houska et al. [74, 75].

Although the basic idea of these approaches, i.e., to derive tight bounds on the discretization error, is similar to our investigation, it has to the best of our knowledge not been studied before if specific methods already produce lower or upper bounds.

The class of continuous-time enclosure techniques is based on results for differential inequalities; see Walter [153]. Using these results auxiliary ODE systems are derived such that their solutions provide lower and upper bounds for the original ODE solution. Singer and Barton [140] construct such an auxiliary system for parametric ODEs as follows. Using a linearization of the right-hand side at a solution for a particular (continuous) parameter, two new right-hand sides are constructed which under- and overestimate the original right-hand side. Building on the results by Walter [153], the solution of the constructed auxiliary system encloses the original solution for all parameters. Moreover, the lower and upper bounds are convex, respectively, concave in the parameters. The construction the auxiliary ODE system has been improved such that the solution yields tighter bounds by using McCormick relaxations by Scott et al. [136] and Scott and Barton [134]. Furthermore, Scott and Barton [133] used these ideas to derive convex and concave relaxations for solutions of differential algebraic equations. Besides, Harwood and Barton [62, 63] derive auxiliary ODE systems such that the solutions are affine linear functions in the parameters for every point in time.

In two articles Villanueva and coworkers [23, 152] provide a framework to combine the two approaches, i.e., the time-discretization techniques and the continuous-time enclosure techniques.

Both techniques can be used for global optimization of ODE constrained problems, however, they do not fit in with our ideas discussed in Section 1.2. For the time-discretization techniques we would have to add variables corresponding to the discretization and are thus limited to the accuracy of the initial discretization. On the other hand, we do not have to add such variables for the continuous-time enclosure techniques, but these approaches require to solve of the auxiliary system with arbitrary precision. Thus, we would lose the desired adaptivity of the discretization in our approach.

2.2 Bounding Scalar ODEs

In this section, we consider scalar parameter-dependent initial value problems

$$y(0) = y^0, \quad \partial_s y(s) = f(s, x, y(s)), \quad s \in [0, S], \tag{2.2}$$

i.e., the solutions are one-dimensional functions $y\colon [0, S] \to \mathbb{R}$. Note that we assume that f is continuously differentiable and that there exists a (unique) solution of this differential equation for all initial values $y^0 \in Y^0 \subseteq \mathbb{R}$ and all parameters $x \in X \subseteq \mathbb{R}^k$, where Y^0 and X are bounded and convex.

In Section 1.2 and above we discussed an idea how to obtain relaxations of the feasible set defined by ODE constraints by using one-step methods for solving the ODEs. Hence, given a discretization $0 = s_0 < s_1 < \cdots < s_N = S$ with $h_i := s_i - s_{i-1}$ for all $i \in [N] = \{1, \ldots, N\}$, we investigate (possibly implicit) one-step methods that can be written in the form

$$y_0 = y^0, \quad y_i = y_{i-1} + h_i\, f_h(s_{i-1}, h_i, x, y_{i-1}, y_i) \quad \forall i \in [N], \tag{2.3}$$

where f_h is the *increment function.* Our goal is to derive sufficient conditions on the increment function f_h such that the method provably yields a lower or upper bound on the true solution of (2.2), i.e., that either $y_N \le y(S)$ or $y(S) \le y_N$ holds. Moreover, we are interested in the input-output functions of particular one-step methods, that is, the functions $(x, y^0, N) \mapsto y_N$ which map the parameters x and initial value y^0 to the last approximation y_N produced through evaluating the one-step method on a discretization with $N + 1$ grid points.

Remark 2.1. Note that in the literature cited above as well as in textbooks on numerical methods for ODEs, see, e.g., Mattheij and Molenaar [97], it is custom to consider ODEs in time. However, in our recurring example of stationary gas transport we consider differential equations in space. Moreover, in mixed-integer nonlinear programming abstract problem settings usually contain variables x. Thus, we decided to consider differential equations in s (which can denote either space or time) and parameters x in this section and the next chapter; see the abstract optimization problem (3.1). Though, in the context of stationary gas transport x denotes the spatial variable. Even so, it should be clear by context in which respect x is used.

In the textbook context of one-step methods for ordinary differential equations, the *global discretization error* $e_i = y(s_i) - y_i$ for $i \in [N]$ is usually studied to prove convergence to the exact solution. By definition the global discretization error might be a good indicator, if a particular method defines lower or upper bounds on the exact solution. However, known techniques, e.g., see Skeel [143] or Lang and Verwer [85], only yield estimates of e_i. Even though, these estimates can be very accurate for small step sizes, they do not a priori prove if the global error is positive or negative. This holds especially for large step sizes. Instead, we will see that the *local discretization error*

$$\tau(s,h) = y(s+h) - y(s) - h\, f_h\big(s,h,x,y(s),y(s+h)\big),$$

i.e., the error produced by plugging the analytical solution of the ODE into the recursion formula (2.3) of the one-step method, provides a suitable indicator.

Example 2.2. We consider an explicit method, that is, $f_h(s,h,x,y,\tilde{y})$ is independent of $\tilde{y}$, with nonnegative local discretization error, i.e.,

$$y(s_i) - y(s_{i-1}) - h_i\, f_h\big(s_{i-1},h_i,x,y(s_{i-1})\big) \geq 0 \tag{2.4}$$

holds for all $i \in [N]$. From (2.4), we immediately get

$$y(s_1) - y_1 = y(s_1) - y(s_0) - h_1\, f_h\big(s_0,h_1,x,y(s_0)\big) \geq 0.$$

Nevertheless, this does not guarantee that $y(s_2) - y_2 \geq 0$ holds. For example, consider the ODE $\partial_s y(s) = -y(s)$ with $y(0) = 1$. For this particular ODE with the solution $y(s) = e^{-s}$, the explicit Euler method given by the increment function $f_h(s_{i-1},h_i,x,y_{i-1},y_i) = f(s_{i-1},y_{i-1})$ with equidistant step size h has a nonnegative local discretization error and produces the solution $y_i = (1-h)^i$ for all i. Thus, with $h = 2$ we have $-1 = y_{2i-1} \leq y(s_{2i-1})$ and $y(s_{2i}) \leq y_{2i} = 1$ for all i. On the other hand, let $0 < h \leq 1$. Since $y(s) = e^{-s} = \lim_{n\to\infty}(1-\frac{s}{n})^n$ holds and $(1-\frac{s}{n})^n$ is increasing in n for $n \geq s$, we can derive that the inequality

$$y(s_i) = e^{-ih} = \lim_{n\to\infty}\big(1-\tfrac{ih}{n}\big)^n \geq \big(1-\tfrac{ih}{i}\big)^i = (1-h)^i = y_i$$

is true for all i.

This example suggests that a signed local discretization error and small step sizes are sufficient for producing lower and upper bounds. In fact, by (2.4)

$$y(s_i) \geq y(s_{i-1}) + h_i\, f_h\big(s_{i-1},h_i,x,y(s_{i-1})\big) \geq y_{i-1} + h_i\, f_h\big(s_{i-1},h_i,x,y_{i-1}\big) = y_i$$

holds if $y(s_{i-1}) \geq y_{i-1}$ and $y + h\, f_h(s,h,x,y)$ is nondecreasing w.r.t. y, which is typically true for small step sizes. Thus, we can derive $y(s_i) \geq y_i$.

Lemma 2.3. *Consider a one-step method of the form* (2.3) *for a scalar ODE* (2.2)*, i.e.,* $y(s) \in \mathbb{R}$*. Let the local discretization error of the method be nonnegative, i.e., for all* $s \in [0, S]$ *and* $h \geq 0$ *with* $s + h \leq S$ *and all parameters* $x \in X$ *the inequality*

$$\tau(s, h) = y(s + h) - y(s) - h\, f_h\big(s, h, x, y(s), y(s + h)\big) \geq 0$$

holds. Suppose the derivatives of the increment function f_h *satisfy*

$$\partial_y f_h(s, h, x, y, \tilde{y}) \geq b \quad \text{and} \quad \partial_{\tilde{y}} f_h(s, h, x, y, \tilde{y}) \leq B$$

for constants b, $B \in \mathbb{R}$*. Then if the step sizes* h_i *satisfy*

$$0 < h_i \leq h_{max} = \begin{cases} \infty, & \text{if } b \geq 0 \text{ and } B \leq 0, \\ \frac{1}{\max\{-b, B\}}, & \text{otherwise}, \end{cases}$$

for all $i \in [N]$*, the one-step method produces a lower bound on the ODE solution* $y(s)$ *for all* $x \in X$*, that is,* $y_i \leq y(s_i)$ *for all* $i \in [N]$*.*

Otherwise, if the local discretization error of the method is nonpositive, i.e.,

$$\tau(s, h) = y(s + h) - y(s) - h\, f_h\big(s, h, x, y(s), y(s + h)\big) \leq 0$$

holds for all $s \in [0, S]$ *and* $h \geq 0$ *with* $s + h \leq S$ *and all parameters* $x \in X$*, then we obtain under the same assumptions upper bounds* $y_i \geq y(s_i)$ *for all* $i \in [N]$*.*

Proof. We prove the lemma by induction on the number of grid points N. Consider the function

$$R(s, h, x, y, \tilde{y}) = \tilde{y} - y - h\, f_h(s, h, x, y, \tilde{y}).$$

Then a step of (2.3) is given by the equation $R(s_{i-1}, h_i, x, y_{i-1}, y_i) = 0$. By the assumption on the derivatives of f_h, we get

$$\partial_{\tilde{y}} R(s, h, x, y, \tilde{y}) = 1 - h\, \partial_{\tilde{y}} f_h(s, h, x, y, \tilde{y}) \geq 1 - h\, B.$$

Obviously, R is nondecreasing w.r.t. $\tilde{y}$ if $B \leq 0$ holds, or if the step size satisfies the inequality $h \leq \frac{1}{B}$. Since the local discretization error is nonnegative and $y_0 = y(0)$, we can derive the inequality

$$R\big(0, h_1, x, y(0), y(s_1)\big) \geq 0 = R\big(0, h_1, x, y_0, y_1\big).$$

Therefore, we obtain the inequality $y(s_1) \geq y_1$ if either $B \leq 0$ or $h_1 \leq \frac{1}{B}$ holds. For induction we assume that $y(s_i) \geq y_i$ is satisfied. Since $\partial_y f_h$ is bounded below by b, we know that

$$\partial_y R(s,h,x,y,\tilde{y}) = -1 - h\,\partial_y f_h(s,h,x,y,\tilde{y}) \leq -1 - h\,b$$

holds. Thus, R is nonincreasing w.r.t. y if either $b \geq 0$ or $h \leq -\frac{1}{b}$ is true. Again, using that the local discretization error is nonnegative we derive

$$R\big(s_i, h_{i+1}, x, y(s_i), y(s_{i+1})\big) \geq 0 = R\big(s_i, h_{i+1}, x, y_i, y_{i+1}\big).$$

Furthermore, the monotonicity w.r.t. the fourth argument and $y(s_i) \geq y_i$ results in

$$R\big(s_i, h_{i+1}, x, y_i, y(s_{i+1})\big) \geq R\big(s_i, h_{i+1}, x, y(s_i), y(s_{i+1})\big) \geq R\big(s_i, h_{i+1}, x, y_i, y_{i+1}\big),$$

and consequently $y(s_{i+1}) \geq y_{i+1}$ holds. Then induction yields that the one-step method produces a lower bound on $y(S)$.

The case of nonpositive local discretization error and the obtained upper bound can be treated in the same way. □

Note that for most one-step methods like Runge-Kutta methods, the assumption that the right-hand side f of the ODE is Lipschitz continuous already ensures that the partial derivatives $\partial_y f_h$ and $\partial_{\tilde{y}} f_h$ are bounded.

Remark 2.4. If we consider an explicit one-step method, i.e., $f_h(s,h,x,y,\tilde{y})$ is independent of $\tilde{y}$ and we get $y_i = y_{i-1} - h_i\,f_h(s_{i-1}, h_i, x, y_{i-1})$, then the previous lemma yields that we can choose

$$h_{\max} = \begin{cases} \infty, & \text{if } b \geq 0, \\ -\frac{1}{b}, & \text{else.} \end{cases}$$

Remark 2.5. If we consider an "end value problem" instead of an initial value problem, that is, $\partial_s y(s) = f(s,x,y(s))$ holds for $s \in [0,S]$ and $y(S) = y^S$, then Lemma 2.3 shows that we can still compute lower and upper bounds on the solution, however, with the modification that the bounds are now reversed, i.e., nonnegative local discretization error now yields upper bounds and nonpositive local discretization error now yields lower bounds.

Remark 2.6. In the autonomous case (f is independent of s), the sign condition of Lemma 2.3 for the local discretization error is also necessary in the following sense.

If there exist s and $y(s)$, such that there is no $\bar{h} > 0$ for which (2.4) holds for all $0 < h \leq \bar{h}$, then the method (2.3) does not produce lower bounds for all initial values y^0. In fact, in the autonomous case without restrictions on the initial value y^0, we can choose $y(0) = y^0$ such that there exists $h_0 > 0$ with nonpositive local discretization error. Then a strict version of the second assertion of Lemma 2.3 is applicable, yielding $y_1 > y(s_1)$, i.e., we obtain an upper instead of a lower bound.

With the sufficient criterion given by Lemma 2.3 at hand, we will characterize conditions under which three particular methods produce lower or upper bounds. Moreover, we study sufficient conditions such that the corresponding input-output functions are convex or concave. Again, let $0 = s_0 < s_1 < \cdots < s_N = S$ be a given discretization with step sizes $h_i = s_i - s_{i-1}$ for all $i \in [N]$. The three methods are the *explicit midpoint method*, the *second-order Taylor method* and the *implicit trapezoidal rule* defined by their respective increment functions

$$f_h^{em}(s,h,x,y) \coloneqq f\big(s + \tfrac{h}{2}, x, y + \tfrac{h}{2} f(s,x,y)\big), \tag{2.5}$$

$$f_h^{ta}(s,h,x,y) \coloneqq f(s,x,y) + \tfrac{h}{2}\big(\partial_y f \cdot f + \partial_s f\big)(s,x,y) \tag{2.6}$$

and

$$f_h^{tr}(s,h,x,y,\tilde{y}) \coloneqq \tfrac{1}{2} f(s,x,y) + \tfrac{1}{2} f(s,x,\tilde{y}). \tag{2.7}$$

Moreover, we denote with $F^{em} \colon X \times Y^0 \times \mathbb{N}$, $F^{ta} \colon X \times Y^0 \times \mathbb{N}$ and $F^{tr} \colon X \times Y^0 \times \mathbb{N}$ the corresponding input-output functions $(x, y^0, N) \mapsto y_N$ which are defined through evaluating the three methods.

Remark 2.7. We point out that we do not study these three methods by chance. Neither are they the only methods Lemma 2.3 can be successfully applied to. The reason to investigate them is that they *do the trick* for the application on stationary gas transport, i.e., they define lower and upper bounds on the solution of (1.11). In the case $\sigma = 0$, lower and upper bounds are given by the explicit midpoint method and the trapezoidal rule. In the case $\sigma \neq 0$, lower and upper bounds can be computed by the second-order Taylor method and the trapezoidal rule. Furthermore, note that also our implementation for the case $\sigma = 0$ is based on the explicit midpoint method and the trapezoidal rule; see the following Sections 2.3, 2.4 and 4.3.

Remark 2.8. An important property for the spatial branch-and-bound approach which we develop in Chapter 3 is that these three methods are convergent under mild assumptions on f; e.g., see Theorem 3.4 and Corollary 3.5 in Chapter 3 of Mattheij and Molenaar [97].

A one-step is called *consistent* if the local discretization error satisfies

$$\lim_{h \to 0} \left| \frac{\tau(s,h)}{h} \right| = 0.$$

Moreover, the method is called *convergent* if the global discretization error satisfies

$$\lim_{h\to 0} \; e_i = \lim_{h\to 0} y(s_i) - y_i = 0$$

for all $i \in [N]$. A method is called *consistent of order k* if $\tau(x,h)h^{-1} \in \mathcal{O}(h^k)$ holds and *convergent of order k* if $e_i \in \mathcal{O}(h^k)$ holds for all $i \in [N]$. Consistency and convergence of a one-step method are related as follows.

Suppose that $y(s) \in Y \subseteq \mathbb{R}$ for all initial values $y^0 \in Y^0$ and parameters $x \in X$. Let f be locally Lipschitz continuous w.r.t. y on $[0,S] \times X \times Y$. Then a one-step method with $y_i \in Y$ for all $i \in [N]$ is convergent if and only if it is consistent. Moreover, the convergence order coincides with the consistency order.

Note that under these assumptions the three methods above are convergent of order 2. To see this we have to show that the local discretization error is consistent of order 2. For example, consider the explicit midpoint method. By Taylor's theorem we get

$$y(s+h) = y(s) + h\,\partial_s y(s) + \tfrac{1}{2}h^2\,\partial_{ss} y(s) + \mathcal{O}(h^3)$$

and

$$\begin{aligned}
&f\big(s + \tfrac{h}{2},\, x, y(s) + \tfrac{h}{2} f(s,x,y(s))\big) \\
&\quad = f(s,x,y(s)) + \tfrac{1}{2}h\,\partial_s f(s,x,y(s)) + \tfrac{1}{2}h\,\partial_y f(s,x,y(s)) \cdot f(s,x,y(s)) + \mathcal{O}(h^2) \\
&\quad = \partial_s y(s) + \tfrac{1}{2}h\,\partial_{ss} y(s) + \mathcal{O}(h^2).
\end{aligned}$$

Together these two equations yield

$$\tau(s,h) = y(s+h) - y(s) - h\,f\big(s + \tfrac{h}{2}, x, y(s) + \tfrac{h}{2} f(s,x,y(s))\big) = \mathcal{O}(h^3),$$

i.e., the explicit midpoint method is consistent and convergent of order 2. Furthermore, that the second-order Taylor method and the trapezoidal rule too have convergence order 2 can be seen analogously.

Explicit Midpoint Method

Direct application of Lemma 2.3 to the explicit midpoint method

$$y_0 = y^0, \quad y_i = y_{i-1} + h_i\, f_h^{em}(s_{i-1}, h_i, x, y_{i-1}) \quad \forall i \in [N], \tag{2.8}$$

with $f_h^{em}(s,h,x,y) = f(s + \frac{h}{2}, x, y + \frac{h}{2} f(s,x,y))$ defined in (2.5) yields the following result.

Corollary 2.9. *Let $f(s,x,y)$ be continuously differentiable and $\partial_y f(s,x,y)$ be non-negative. If both $y(s)$ and $\partial_s y(s)$ are convex then the explicit midpoint method pro-*

duces lower bounds on $y(s_i)$ for all $i \in [N]$. If both $y(s)$ and $\partial_s y(s)$ are concave then the method produces upper bounds.

Otherwise, let $\partial_y f(s,x,y)$ be nonpositive and bounded, i.e., $b \le \partial_y f(s,x,y) \le 0$ holds for some $b \in \mathbb{R}$, and let the step sizes h_i satisfy the condition $0 < h_i \le -\frac{1}{b}$ for all $i \in [N]$. Then y_i is a lower bound on the ODE solution $y(s_i)$ for all $i \in [N]$ if $y(s)$ is concave and $\partial_s y(s)$ is convex. On the other hand, y_i is an upper bound if $y(s)$ is convex and $\partial_s y(s)$ is concave.

Proof. Let $\partial_y f(s,x,y)$ be nonnegative and $y(s)$ as well as $\partial_s y(s)$ be convex. In order to use Lemma 2.3 we have to show that the local discretization error

$$\tau(s,h) = y(s+h) - y(s) - h\, f\big(s + \tfrac{h}{2}, x, y(s) + \tfrac{h}{2} f(s,x,y(s))\big)$$

is nonnegative.

Because $\partial_s y(s)$ is convex, we obtain

$$\begin{aligned} y(s+h) - y(s) &= \int_s^{s+h} \partial_s y(\tilde{s})\, \mathrm{d}\tilde{s} \\ &\ge \int_s^{s+h} \partial_s y(s+\tfrac{h}{2}) + \partial_{ss} y(s+\tfrac{h}{2})\big(\tilde{s} - (s+\tfrac{h}{2})\big)\, \mathrm{d}\tilde{s} \qquad (2.9) \\ &= h\, \partial_s y(s+\tfrac{h}{2}) + \partial_{ss} y(s+\tfrac{h}{2})\big[\tfrac{1}{2}(s+h)^2 - \tfrac{1}{2}s^2 - (s+\tfrac{h}{2})h\big] \\ &= h\, f\big(s+\tfrac{h}{2}, x, y(s+\tfrac{h}{2})\big). \end{aligned}$$

Moreover, by assumption $y(s)$ is convex, too. Hence the inequality

$$y(s+\tfrac{h}{2}) \ge y(s) + \tfrac{h}{2}\partial_s y(s) \qquad (2.10)$$

holds. Together with $\partial_y f(s,x,y) \ge 0$ the inequalities (2.9) and (2.10) yield

$$y(s+h) - y(s) - h\, f\big(s+\tfrac{h}{2}, x, y(s) + \tfrac{h}{2} f(s,x,y(s))\big) \ge 0,$$

and thus by Lemma 2.3 the explicit midpoint method produces lower bounds on the solution of the ODE. Furthermore, the other cases are similar. □

Note that one does not have to know the exact solution of an ODE to determine whether the assumptions of Corollary 2.9 are satisfied. Since the second derivative of the solution is given by

$$\partial_{ss} y(s) = \partial_s f\big(s,x,y(s)\big) + \partial_y f\big(s,x,y(s)\big) \cdot f\big(s,x,y(s)\big), \qquad (2.11)$$

it often suffices to analyze the right-hand side to check whether the solution $y(s)$ is convex or concave. Analogously, we can check the condition on $\partial_s y(s)$.

Corollary 2.9 shows that in particular cases we can use the explicit midpoint method to define a lower or upper bound on the solution of an ODE constraint; see Section 1.2. For the use in a spatial branch-and-bound algorithm it is desirable that relaxations are either convex or concave, thus we state sufficient conditions such that the input-output function $F^{em}\colon (x, y_0, N) \mapsto y_N$ is convex or concave.

Lemma 2.10. *Let $f(s,x,y)$ be continuously differentiable and convex in (x,y) for all $s \in [0,S]$, and let $\partial_y f(s,x,y)$ be nonnegative. Then the input-output function $F^{em}\colon X \times Y^0 \times \mathbb{N} \to \mathbb{R}$ defined by $(x, y^0, N) \mapsto y_N$ through evaluating the explicit midpoint method* (2.8) *with parameter x, initial value y^0 and N discretization steps is continuously differentiable and convex in (x, y^0).*

If $f(s,x,y)$ is concave instead of convex, then F^{em} is concave w.r.t. (x, y^0).

Proof. We consider the function which describes a single step of the explicit midpoint method, that is,

$$y^{em}(s,h,x,y) = y + h\, f\big(s + \tfrac{h}{2}, x, y + \tfrac{h}{2} f(s,x,y)\big).$$

Then we can write $y_i = y^{em}(s_{i-1}, h_i, x, y_{i-1})$ for all $i \in [N]$.

Let $y, y' \in Y^0$ and $x, x' \in X$, and denote their convex combinations with $\mu \in (0,1)$ by $\tilde{y} = \mu\, y + (1-\mu)y'$ and $\tilde{x} = \mu\, x + (1-\mu)x'$. Since f is by assumption convex in (x,y), the inequality

$$\tilde{y} + \tfrac{h}{2}\, f(s,\tilde{x},\tilde{y}) \le \mu[y + \tfrac{h}{2}\, f(s,x,y)] + (1-\mu)[y' + \tfrac{h}{2}\, f(s,x',y')]$$

holds. Together with $\partial_y f(s,x,y)$ being nonnegative, we can derive

$$\begin{aligned}
y^{em}(s,h,\tilde{x},\tilde{y}) &= \tilde{y} + h\, f\big(s + \tfrac{h}{2}, \tilde{x}, \tilde{y} + \tfrac{h}{2} f(s,\tilde{x},\tilde{y})\big) \\
&\le \tilde{y} + h\, f\big(s + \tfrac{h}{2}, \tilde{x}, \mu[y + \tfrac{h}{2} f(s,x,y)] + (1-\mu)[y' + \tfrac{h}{2} f(s,x',y')]\big) \\
&\le \tilde{y} + \mu\, h\, f\big(s + \tfrac{h}{2}, x, y + \tfrac{h}{2} f(s,x,y)\big) + (1-\mu) h\, f\big(s + \tfrac{h}{2}, x', y' + \tfrac{h}{2} f(s,x',y')\big) \\
&= \mu\, y^{em}(s,h,x,y) + (1-\mu)\, y^{em}(s,h,x',y').
\end{aligned}$$

Hence, y^{em} is convex w.r.t. (x,y). Additionally, y^{em} is continuously differentiable if f is continuously differentiable and, since $\partial_y f(s,x,y)$ is nonnegative, we can derive that $\partial_y y^{em}(s,h,x,y) \ge 1$ is true.

Since the composition of a nondecreasing convex function with a convex function is convex and y^{em} is continuously differentiable, we can inductively show that y_N

is continuously differentiable and convex w.r.t. parameter and initial value. Analogously, we can see that it is concave if f is concave. □

Note that in the case of $\partial_y f(s,x,y) \leq 0$ the proof above does not work, but y^{em} and also F^{em} might still be convex or concave. For example, in Section 2.3 we are going to apply the explicit midpoint method to the differential equation (1.11) and see that the input-output function is convex although the derivative of the right-hand side is nonpositive.

Second-Order Taylor Method

Since the assumptions of Corollary 2.9 are rather strict, we investigate the second-order Taylor method

$$y_0 = y^0, \quad y_i = y_{i-1} + h_i\, f_h^{ta}(s_{i-1}, h_i, x, y_{i-1}) \quad \forall i \in [N] \tag{2.12}$$

with $f_h^{ta}(s,h,x,y) = f(s,x,y) + \frac{h}{2}(\partial_y f \cdot f + \partial_s f)(s,x,y)$ as defined in (2.6). The investigation of this method is especially motivated by the application to gas flow in pipelines with positive slope; see Section 2.4. Again, by using Lemma 2.3 we can derive the following result analogously to Corollary 2.9.

Corollary 2.11. *Let $f(s,x,y)$ be twice continuously differentiable. Let b, $b' \in \mathbb{R}$ with $b \leq 0$ and $b' \geq 0$, and suppose that*

$$b \leq \partial_y f(s,x,y) \quad \text{and} \quad -b' \leq \big(\partial_{yy} f \cdot f + (\partial_y f)^2 + \partial_{sy} f\big)(s,x,y)$$

holds for all $x \in X$. Furthermore, let $\partial_s y(s)$ be convex. The second-order Taylor method (2.12) produces lower bounds $y_i \leq y(s_i)$ for all $i \in [N]$ if the step sizes h_i satisfy the inequality

$$0 < h_i \leq \begin{cases} \infty & \text{if } b = b' = 0, \\ \frac{2}{-b+\sqrt{b^2+2b'}} & \text{otherwise.} \end{cases} \tag{2.13}$$

If $\partial_s y(s)$ is concave, then under the above condition on the step sizes h_i the second-order Taylor method (2.12) produces upper bounds on $y(s_i)$ for all $i \in [N]$.

Proof. Suppose that $\partial_s y(s)$ is convex. Then using equation (2.11) we can derive

$$\begin{aligned} y(s+h) - y(s) &= \int_s^{s+h} \partial_s y(\tilde{s})\,\mathrm{d}\tilde{s} \geq \int_s^{s+h} \partial_s y(s) + \partial_{ss} y(s) \cdot (\tilde{s} - s)\,\mathrm{d}\tilde{s} \\ &= h\, f_h^{ta}(s,h,x,y(s)). \end{aligned}$$

Thus, the second-order Taylor method has a nonnegative local discretization error. Otherwise, if $\partial_s y(s)$ is concave, we can analogously show that the local discretization error is nonpositive.

In order to apply Lemma 2.3, we additionally have to show that $\partial_y f_h^{ta}$ is bounded from below. Since

$$\partial_y f_h^{ta}(s,h,x,y) = \partial_y f(s,x,y) + \tfrac{h}{2}\big(\partial_{yy} f \cdot f + (\partial_y f)^2 + \partial_{sy} f\big)(s,x,y) \geq b - \tfrac{h}{2}\, b',$$

the derivative $\partial_y f_h^{ta}$ is nonnegative if $b = b' = 0$ holds. Otherwise, choosing h according to (2.13) implies

$$\partial_y f_h^{ta}(s,h,x,y) \geq b - \tfrac{h}{2}\, b' \geq b + \tfrac{1}{b-\sqrt{b^2+2b'}}\, b' = b + \tfrac{b+\sqrt{b^2+2b'}}{b^2-(b^2+2b')}\, b' = \tfrac{b-\sqrt{b^2+2b'}}{2},$$

i.e., $\partial_y f_h^{ta}$ is bounded from below. Hence, the second-order Taylor method produces lower bounds if $\partial_s y(s)$ is convex and upper bounds if $\partial_s y(s)$ is concave. $\square$

Note that in Corollary 2.9 we assumed (in the case $\partial_y f \geq 0$) that the solution $y(s)$ and its derivative $\partial_s y(s)$ are convex to obtain lower bounds through the explicit midpoint method. Whereas in Corollary 2.11 we only assumed the convexity of $\partial_s y(s)$ (besides of the boundedness of the increment function) to obtain lower bounds through the second-order Taylor method. In this sense, the assumptions here are less strict.

We now state conditions that ensure the convexity or concavity of the input-output function $F^{ta}\colon (x, y^0, N) \mapsto y_N$ defined by the second-order Taylor method.

Lemma 2.12. *Let f be three times continuously differentiable and let b, $b' \in \mathbb{R}$ with $b \leq 0$ and $b' \geq 0$. Suppose that*

$$b \leq \partial_y f(s,x,y) \quad \textit{and} \quad -b' \leq \big(\partial_{yy} f \cdot f + (\partial_y f)^2 + \partial_{sy} f\big)(s,x,y)$$

holds for all $x \in X$. Assume that there exists $\bar{h} > 0$ satisfying (2.13) such that the increment function f_h^{ta} is convex w.r.t. (x,y) for all $s \in [0,S]$ and $0 < h \leq \bar{h}$. Then with $0 < h_i \leq \bar{h}$ for all $i \in [N]$ the input-output function $F^{ta}\colon X \times Y^0 \times \mathbb{N} \to \mathbb{R}$ defined through evaluating the second-order Taylor method (2.12) is continuously differentiable and convex w.r.t. (x, y^0).

Alternatively, let $f(s,x,y)$ be uniformly convex w.r.t. (x,y) and its derivatives up to order three be bounded. Then there exists $\bar{h} > 0$ such that F^{ta} is convex w.r.t. (x, y^0) if $0 < h_i \leq \bar{h}$ for all $i \in [N]$.

If f_h^{ta} is concave or $f(s,x,y)$ is uniformly concave instead of convex, then F^{ta} is concave w.r.t. (x, y^0) under the remaining assumptions above.

Proof. Consider the function $y^{ta}(s,h,x,y) \coloneqq y + h\, f_h^{ta}(s,h,x,y)$ defining one step of the second-order Taylor method (2.12). Then we can write (2.12) as

$$y_0 = y^0, \quad y_i = y^{ta}(s_{i-1}, h_i, x, y_{i-1}) \quad \forall i \in [N].$$

Hence, if y^{ta} is nondecreasing w.r.t. y and convex w.r.t. (x,y), we can inductively derive that F^{ta} is convex w.r.t. (x,y).

By assumption there exists $\bar{h} > 0$ satisfying (2.13) such that the increment function f_h^{ta} is convex for $0 < h \leq \bar{h}$. Then $y^{ta}(s,h,x,y)$ is convex. Moreover, as shown in Corollary 2.11, the inequality $\partial_y f_h^{ta}(s,h,x,y) \geq \frac{b-\sqrt{b^2+2b'}}{2}$ holds if the step size h satisfies (2.13). Hence, (2.13) also implies $\partial_y y^{ta}(s,h,x,y) \geq 0$.

In the alternative case, assume that $f(s,x,y)$ is uniformly convex w.r.t. (x,y). Then the Hessian matrix $\nabla^2 f(s,x,y)$ w.r.t. x and y only is uniformly positive definite. Moreover, by assumption the derivatives up to order three of $f(s,x,y)$ are bounded, hence there exist b, b' satisfying the assumptions of the first case. Then (2.13) ensures that we have $\partial_y y^{ta}(s,h,x,y) \geq 0$.

For the Hessian matrix $\nabla^2 y^{ta}(s,h,x,y)$ w.r.t. x and y only we have

$$\nabla^2 y^{ta}(s,h,x,y) = h\nabla^2 f(s,y,x) + \tfrac{h^2}{2}\nabla^2(\partial_y f \cdot f + \partial_s f)(s,x,y).$$

Since $\nabla^2 f(s,y,x)$ is uniformly positive definite and the derivatives in the second term are bounded, we can choose $\bar{h}$ satisfying (2.13) small enough such that $\nabla^2 y^{ta}$ is positive definite for all $0 < h \leq \bar{h}$. Then $y^{ta}(s,h,x,y)$ is convex in (x,y) and nondecreasing w.r.t. y and thus F^{ta} is convex w.r.t. (x,y).

Finally, the cases where f_h^{ta} is concave or $f(s,x,y)$ is uniformly concave instead of convex can be handled analogously. $\square$

Trapezoidal Rule

To obtain bounds opposite to the bounds produced by either the explicit midpoint method or the second-order Taylor method, e.g., upper bounds in cases where the other methods define lower bounds, we consider the implicit trapezoidal rule

$$y_0 = y^0, \quad y_i = y_{i-1} + h_i\, f_h^{tr}(s_{i-1}, h_i, x, y_{i-1}, y_i) \quad \forall i \in [N] \tag{2.14}$$

with $f_h^{tr}(s,h,x,y,\tilde{y}) \coloneqq \frac{1}{2}\, f(s,x,y) + \frac{1}{2}\, f(s,x,\tilde{y})$ as defined in (2.7). Again, by using Lemma 2.3 we can derive the following corollary.

Corollary 2.13. *Let $\partial_s y(s)$ be convex, $f(s,x,y)$ be continuously differentiable, and the derivative $b \leq \partial_y f(s,x,y) \leq B$ be bounded by some constants b, $B \in \mathbb{R}$ for all $x \in X$. Furthermore, suppose that for all $i \in [N]$ the step sizes h_i satisfy the* *condition* $h_i \cdot \max\{-b, B\} \leq 2$ *and a solution* y_i *of* (2.14) *exists. Then* y_i *is an upper bound on the ODE solution* $y(s_i)$ *for all* $i \in [N]$.

If $\partial_s y(s)$ *is concave instead of convex, then* y_i *is a lower bound for all* $i \in [N]$.

Proof. We only discuss the case in which $\partial_s y(s)$ is convex, the other case works analogously. By Lemma 2.3 we have to show that the local discretization error is nonpositive. Since $\partial_s y(s) = f\big(s, x, y(s)\big)$ is convex, the inequality

$$f\big(\tilde{s}, x, y(\tilde{s})\big) \leq f\big(s, x, y(s)\big) + \tfrac{1}{h}\big[f\big(s+h, x, y(s+h)\big) - f\big(s, x, y(s)\big)\big](\tilde{s} - s)$$

holds for all $\tilde{s} \in [s, s+h]$. With this we can derive the inequality

$$\begin{aligned} y\big(s+h\big) - y\big(s\big) &= \int_s^{s+h} f\big(\tilde{s}, x, y(\tilde{s})\big)\,\mathrm{d}\tilde{s} \\ &\leq \int_s^{s+h} f\big(s, x, y(s)\big) + \tfrac{1}{h}\big[f\big(s+h, y(s+h)\big) - f\big(s, x, y(s)\big)\big](\tilde{s} - s)\,\mathrm{d}\tilde{s} \\ &= \tfrac{h}{2}\big[f\big(s, x, y(s)\big) + f\big(s+h, x, y(s+h)\big)\big] \end{aligned}$$

and thus the local discretization error is nonpositive.

By assumption the derivative $\partial_y f(s, x, y)$ is bounded from below by b and from above by B, which implies

$$\partial_y f_h^{tr}(s, h, x, y, \tilde{y}) = \tfrac{1}{2}\partial_y f(s, x, y) \geq \tfrac{b}{2}$$

and

$$\partial_{\tilde{y}} f_h^{tr}(s, h, x, y, \tilde{y}) = \tfrac{1}{2}\partial_{\tilde{y}} f(s, x, \tilde{y}) \leq \tfrac{B}{2},$$

i.e., the boundedness conditions of Lemma 2.3 are satisfied. Then, the upper bound on the step sizes immediately follows from Lemma 2.3. Therefore, the trapezoidal rule produces upper bounds on the solution $y(s_i)$ for all $i \in [N]$. □

As in the paragraphs above, we now consider sufficient conditions such that the input-output function $F^{tr}\colon X \times Y^0 \times \mathbb{N}$ defined by evaluating the trapezoidal rule is convex or concave.

Lemma 2.14. *Let* $f(s, x, y)$ *be continuously differentiable and convex in* (x, y) *for all* $s \in [0, S]$, *and let* $b \leq \partial_y f(s, x, y) \leq B$ *for some constants* b, $B \in \mathbb{R}$. *Furthermore, suppose that the step sizes satisfy the condition* $h_i \cdot \max\{-b, B\} < 2$ *and there exists a solution to* (2.14) *for all* $i \in [N]$. *Then the input-output func-*

tion $F^{tr}\colon X \times Y^0 \times \mathbb{N} \to \mathbb{R}$ *defined by* $(x, y^0, N) \mapsto y_N$ *through evaluating the trapezoidal rule* (2.14) *is continuously differentiable and convex in* (x, y^0).

If $f(s, x, y)$ *is concave instead of convex, then* F^{tr} *is concave w.r.t.* (x, y^0).

Proof. We consider the following function, which is defined by a single step of the trapezoidal rule:

$$R(s, h, x, y, \tilde{y}) = \tilde{y} - y - \tfrac{h}{2}\big[f(s, x, y) + f(s + h, x, \tilde{y})\big].$$

By assumption, there exists a solution y_i of $R(s_{i-1}, h_i, x, y_{i-1}, y_i) = 0$ for all $i \in [N]$. Furthermore, the inequality

$$\partial_{\tilde{y}} R(s, h, x, y, \tilde{y}) = 1 - \tfrac{h}{2}\,\partial_y f(s + h, x, \tilde{y}) \geq 1 - \tfrac{h}{2}\,B > 0$$

holds, if $B \leq 0$ or $h < \frac{2}{B}$ is satisfied, i.e., the assumptions of the implicit function theorem are satisfied. Thus, there exists a continuously differentiable function $y^{tr}(s, h, x, y)$ with

$$y^{tr}(s, h, x, y) - \tfrac{h}{2}\,f\big(s + h, x, y^{tr}(s, h, x, y)\big) = y + \tfrac{h}{2}\,f\big(s, x, y\big). \tag{2.15}$$

Analogously to the proof of Lemma 2.10, we will use this function to inductively obtain that y_N is a convex function w.r.t. (x, y^0) assuming that $f(s, x, y)$ is a convex function w.r.t. (x, y).

Let $\tilde{y} = \mu y + \mu' y'$ and $\tilde{x} = \mu x + \mu' x'$ for some $\mu \in (0, 1)$ and $\mu' = 1 - \mu$. By the definition of y^{tr}, we can derive the inequality

$$\begin{aligned}
y^{tr}(s, h, \tilde{x}, \tilde{y}) - \tfrac{h}{2}\,f\big(s + h, \tilde{x}, y^{tr}(s, h, \tilde{x}, \tilde{y})\big)& \\
&= \tilde{y} + \tfrac{h}{2}\,f\big(s, \tilde{x}, \tilde{y}\big) \\
&\leq \tilde{y} + \tfrac{h}{2}\big[\mu\, f(s, x, y) + \mu'\, f(s, x', y')\big] \\
&= \mu\,\big[y^{tr}(s, h, x, y) - \tfrac{h}{2}\,f\big(s + h, x, y^{tr}(s, x, y)\big)\big] \\
&\qquad + \mu'\,\big[y^{tr}(s, h, x', y') - \tfrac{h}{2}\,f\big(s + h, x', y^{tr}(s, h, x', y')\big)\big] \\
&\leq \mu\, y^{tr}(s, x, y) + \mu'\, y^{tr}(s, x', y') \\
&\qquad - \tfrac{h}{2}\,f\big(s + h, \tilde{x}, \mu\, y^{tr}(s, h, x, y) + \mu'\, y^{tr}(s, h, x', y')\big).
\end{aligned}$$

From this we can derive the convexity of $y^{tr}(s, h, x, y)$ if $y - \frac{h}{2}\,f(s, x, y)$ is nondecreasing w.r.t. y for all $x \in X$, that is, if $1 - \frac{h}{2}\partial_y f(s, x, y) \geq 0$ holds. As we have seen before, this is true due to the choice of $h \leq \frac{2}{B}$ if $B > 0$. Thus, $y^{tr}(s, h, x, y)$ is convex.

Next, we show that $y^{tr}(s,h,x,y)$ is nondecreasing w.r.t. y. By differentiating (2.15) we can derive

$$\partial_y y^{tr}(s,h,x,y)\big[1 - \tfrac{h}{2}\,\partial_y f\big(s+h,x,y^{tr}(s,h,x,y)\big)\big] = 1 + \tfrac{h}{2}\,\partial_y f\big(s,x,y\big).$$

The right-hand side is nonnegative if either $b \geq 0$ or $h < \frac{2}{-b}$ is satisfied. Together with the strict inequality $1 - \frac{h}{2}\,\partial_y f(s+h,x,y^{tr}(s,h,x,y)) > 0$ if either $B \leq 0$ or $h < \frac{2}{B}$ holds, this yields that $y^{tr}(s,h,x,y)$ is nondecreasing.

By interpreting the approximations y_i for $i \in [N]$ of the ODE solution as a function of parameter and initial value, we can use the following representation

$$y_i(x,y^0) = y^{tr}\big(s_{i-1},h_i,x,y_{i-1}(x,y^0)\big).$$

Since $y^{tr}(s,h,x,y)$ is continuously differentiable and nondecreasing, and the composition of a convex function with a nondecreasing convex function is convex, we can inductively derive that F^{tr} is continuously differentiable and convex w.r.t. (x,y^0).

Again, the case with a concave right-hand side f can be treated analogously. □

Remark 2.15. Note that the proof of Lemma 2.14 shows that if $B \leq 0$, we can actually choose the step sizes such that $0 < h_i \cdot \max\{-b, 0\} \leq 2$ is satisfied, where in contrast to the assumptions of Lemma 2.14 equality is allowed.

Coming back to the idea on how to relax ODE constraints discussed in Section 1.2 and in the beginning of this chapter. We have seen that in particular cases the explicit midpoint method, the second-order Taylor method and the trapezoidal rule produce lower or upper bounds on the solution of a scalar parameter-dependent ordinary differential equation. Then if two methods produce opposite bounds, we can use the corresponding input-output functions to define a relaxation of the ODE constraint. For example, lower and upper bounds are produced by the explicit midpoint method and the trapezoidal rule if $y(s)$, $\partial_s y(s)$ are convex and $\partial_y f(s,x,y)$ is nonnegative and bounded. Then a relaxation of (2.1) is given by

$$F^{em}(x,y^0,N) \;\leq\; y^S \;\leq\; F^{tr}(x,y^0,N).$$

If additionally $f(s,x,y)$ is convex, F^{em} and F^{tr} are convex too. Hence, we can utilize this to construct an LP-relaxation of this inequality as follows. Since F^{em} is convex and continuously differentiable, we can generate gradient cuts to approximate F^{em} by outer-approximation; see Duran and Grossmann [30]. Moreover, because F^{tr} is convex the so-called concave envelope of F^{tr}, i.e., the smallest concave function which is greater or equal than F^{tr}, can be constructed by using only the values of F^{tr} at the extreme points of $X \times Y^0$ if X and Y^0 are polytopes; e.g., see Horst and Tuy [73, Theorem IV.6].

2.3 Gas Flow in Pipelines without Height Differences

In this section we apply the results for the explicit midpoint method and the trapezoidal rule to gas flow in pipelines without height differences. Recall that we use the stationary isothermal Euler equation (1.11) to describe the gas flow in pipelines. Without height differences, i.e., slope $\sigma = 0$, the differential equation is given by

$$\partial_x p(x) = \varphi\big(p(x), q\big) = -\frac{1}{2}\frac{\lambda c^2 q|q| p(x)}{D(A^2 p^2(x) - c^2 q^2)}, \quad x \in [0, L].$$

Note, we assumed that $\frac{c|q|}{Ap(x)}$ is bounded from above by $\nu_c \in (0,1)$, that is, we have $\frac{c|q|}{Ap(x)} \leq \nu_c$, to derive the ODE in this form. To make sure that this condition holds for all $x \in [0, L]$, we assume that the mass flow q is nonnegative and moreover we consider an "end value problem" instead of an initial value problem, i.e., we consider

$$p(L) = p^0, \quad \partial_x p(x) = \varphi\big(p(x), q\big), \quad x \in [0, L],$$

where we fix the pressure at the end of the pipeline. Since we will shortly see that the pressure is nonincreasing in x, by choosing (p^0, q) in

$$U := \big\{(p, q) \in \mathbb{R}^2 \,:\, 0 < \underline{p} \leq p \leq \overline{p},\ 0 \leq \underline{q} \leq q \leq \overline{q},\ c\, q \leq \nu_c A\, p\big\}$$

the condition $\frac{c|q|}{Ap(x)} \leq \nu_c$ is satisfied for all $x \in [0, L]$. Note that we assume that lower and upper bounds on the pressure (at both ends of the pipeline) and mass flow are given. Moreover, during a branch-and-bound process we can ensure that the mass flow is nonnegative by branching w.r.t. $q = 0$ and reorientation of the pipeline on the branch with $q \leq 0$.

To start our analysis of gas flow, we investigate properties of the right-hand side φ. To this end, we define the domain of φ through

$$U^\varphi := \big\{(p, q) \in \mathbb{R}^2 \,:\, \underline{p} \leq p,\ \underline{q} \leq q \leq \overline{q},\ c\, q \leq \nu_c A\, p\big\}.$$

Note that compared to U defined above, we relax the upper bound on p to ensure $p(x) \in U^\varphi$ for all $x \in [0, L]$. Then differentiating and computing the eigenvalues of the Hessian matrix of φ yields the following result.

Lemma 2.16. *The function $\varphi\colon U^\varphi \to \mathbb{R}$ is nonpositive, nondecreasing in p, nonincreasing in q and concave in $(p,q) \in U^\varphi$. Its second derivatives satisfy $\partial_{pp}\varphi(p,q) \le 0$, $\partial_{pq}\varphi(p,q) \ge 0$ and $\partial_{qq}\varphi(p,q) < 0$. Furthermore, we have that $\varphi(p,q)$, $\partial_p\varphi(p,q)$, $\partial_q\varphi(p,q)$, $\partial_{pp}\varphi(p,q)$ or $\partial_{pq}\varphi(p,q)$ are zero if and only if $q = 0$.*

Proof. On U^φ we have by assumption $q \ge 0$ and $A^2p^2 > c^2q^2$. Thus, φ and its first and second derivatives satisfy

$$\begin{aligned}
\varphi(p,q) &= -\frac{\lambda c^2 q^2 p}{2D(A^2p^2 - c^2q^2)} \le 0,\\
\partial_p\varphi(p,q) &= \frac{\lambda c^2 q^2 (A^2p^2 + c^2q^2)}{2D(A^2p^2 - c^2q^2)^2} \ge 0,\\
\partial_q\varphi(p,q) &= -\frac{\lambda c^2 q A^2 p^3}{D(A^2p^2 - c^2q^2)^2} \le 0,\\
\partial_{pp}\varphi(p,q) &= -\frac{\lambda c^2 q^2 A^2 p (A^2p^2 + 3c^2q^2)}{D(A^2p^2 - c^2q^2)^3} \le 0,\\
\partial_{pq}\varphi(p,q) &= \frac{\lambda c^2 q A^2 p^2 (A^2p^2 + 3c^2q^2)}{D(A^2p^2 - c^2q^2)^3} \ge 0
\end{aligned}$$

and

$$\partial_{qq}\varphi(p,q) = -\frac{\lambda c^2 A^2 p^3 (A^2p^2 + 3c^2q^2)}{D(A^2p^2 - c^2q^2)^3} < 0.$$

Furthermore, the Hessian matrix $\nabla^2\varphi(p,q)$ is singular, that is, the eigenvalues of $\nabla^2\varphi(p,q)$ are 0 and $\partial_{pp}\varphi(p,q)+\partial_{qq}\varphi(p,q)$. Since $\partial_{pp}\varphi(p,q)+\partial_{qq}\varphi(p,q)$ is negative, the Hessian matrix is negative semidefinite, i.e., φ is concave. $\square$

This leads to the following properties of the differential equation.

Corollary 2.17. *The ordinary differential equation*

$$p(L) = p^0, \quad \partial_x p(x) = \varphi\big(p(x), q\big), \quad x \in [0, L] \tag{2.16}$$

has a unique solution $p(x)$ for all $(p^0, q) \in U$. Furthermore, $p(x)$ as well as $\partial_x p(x)$ are nonincreasing and concave. In particular, $p(x) \in U^\varphi$ holds for all $x \in [0, L]$.

Proof. Let $q = 0$. Then for all $(p^0, 0) \in U$, the ODE has the unique solution $p(x) = p^0$, since $\varphi(p, 0) = 0$.

Otherwise, for fixed $q > 0$, the right-hand side $\varphi(p(x), q)$ is negative, i.e., $p(x)$ is decreasing and the pressure is bounded from below by p^0. Thus, since $\partial_{pp}\varphi(p,q) \le 0$

holds, the derivative $\partial_p \varphi(p,q)$ is bounded by $\partial_p \varphi(\frac{c\,q}{\nu_c A}, q) \geq \partial_p \varphi(p,q) > 0$, i.e., φ is Lipschitz continuous w.r.t. p. Hence, the ODE has a unique solution.

Finally, further derivatives of $p(x)$ are given by

$$\begin{aligned}
\partial_{xx} p(x) &= (\partial_p \varphi\, \varphi)(p(x), q),\\
\partial_{xxx} p(x) &= (\partial_{pp} \varphi\, \varphi^2 + (\partial_p \varphi)^2\, \varphi)(p(x), q).
\end{aligned}$$

With the properties of φ derived in Lemma 2.16 this yields that $\partial_{xx} p(x)$ and $\partial_{xxx} p(x)$ are nonpositive, i.e., $p(x)$ as well as $\partial_x p(x)$ are nonincreasing and concave. Moreover, since $p(x)$ is bounded from below by p^0, this implies $p(x) \in U^\varphi$. $\square$

To apply the explicit midpoint method, the trapezoidal rule, and Corollaries 2.9 and 2.13, we transform (2.16) into an initial value problem. To this end, we consider the function $\tilde{p}(x) := p(L - x)$. Then $\tilde{p}(x)$ satisfies the differential equation

$$\tilde{p}(0) = p^0, \quad \partial_x \tilde{p}(x) = -\varphi\big(\tilde{p}(x), q\big), \quad x \in [0, L]. \tag{2.17}$$

For simplicity we consider an equidistant discretization $0 = \tilde{x}_0 < \tilde{x}_1 < \ldots < \tilde{x}_N = L$ with step size $h = \frac{L}{N}$. Then the explicit midpoint method is defined by

$$p_0^\ell = p^0, \qquad p_i^\ell = p_{i-1}^\ell - h\,\varphi\big(p_{i-1}^\ell - \tfrac{h}{2}\varphi(p_{i-1}^\ell, q), q\big) \qquad \forall i \in [N], \tag{2.18}$$

and the implicit trapezoidal rule is

$$p_0^u = p^0, \qquad p_i^u = p_{i-1}^u - \tfrac{h}{2}\left[\varphi\big(p_{i-1}^u, q\big) + \varphi\big(p_i^u, q\big)\right] \qquad \forall i \in [N]. \tag{2.19}$$

In the next step, we deduce by Corollaries 2.9 and 2.13 that the methods define lower and upper bounds on $\tilde{p}(\tilde{x}_i)$ for all $i \in [N]$. Hence, $p_i^\ell \leq p(x_i) \leq p_i^u$ holds with $x_i = L - \tilde{x}_i$ for all $i \in [N]$.

Corollary 2.18. *Let $(p^0, q) \in U$ and N be sufficiently big such that step size $h = \frac{L}{N}$ satisfies the inequality*

$$0 < h \leq \frac{2D}{\lambda} \frac{(1-\nu_c^2)^2}{(1+\nu_c^2)\,\nu_c^2}.$$

Then the explicit midpoint method (2.18) *defines lower bounds on the solution $p(x)$ of* (2.16)*, that is $p_i^\ell \leq p(x_i)$ for all $i \in [N]$.*

For instance, if $\nu_c = 0.4$, then the step size has to satisfy $0 < h \leq \frac{D}{\lambda}\frac{441}{48}$.

Proof. By Corollary 2.17 we can derive that the solution $\tilde{p}(x) = p(L - x)$ of (2.17) is concave and $\partial_x \tilde{p}(x) = -\partial_x p(L - x)$ is convex. Furthermore, the right-hand side satisfies

$$0 \geq -\partial_p \varphi\big(p, q\big) \geq -\partial_p \varphi\big(\tfrac{c\,q}{\nu_c A}, q\big) = -\frac{\lambda}{2D} \frac{(1+\nu_c^2)\,\nu_c^2}{(1-\nu_c^2)^2} =: b$$

for all $(p, q) \in U^\varphi$, since $-\partial_{pp}\varphi(p, q) \geq 0$ holds. Hence, by Corollary 2.9 the explicit midpoint method produces lower bounds $p_i^\ell \leq \tilde{p}(\tilde{x}_i)$ by choosing $0 < h \leq -\frac{1}{b}$. Thus, by definition of $\tilde{p}(x) = p(L - x)$ the explicit midpoint method produces lower bounds p_i^ℓ on $p(x_i)$ for all $i \in [N]$. □

Analogously to this corollary, we can immediately derive the following result by using Corollary 2.13.

Corollary 2.19. *Let $(p^0, q) \in U$ and N sufficiently big such that step size h satisfies*

$$0 < h \leq \frac{4D}{\lambda} \frac{(1-\nu_c^2)^2}{(1+\nu_c^2)\,\nu_c^2}.$$

Then the trapezoidal rule (2.19) *defines upper bounds on the solution $p(x)$ of* (2.16), *that is, the inequality $p_i^u \geq p(x_i)$ holds for all $i \in [N]$.*

For instance, if $\nu_c = 0.4$, then the step size has to satisfy $0 < h \leq \frac{D}{\lambda}\frac{441}{24}$.

Proof. Since the trapezoidal rule is an implicit method, we first show that there exists a solution p_i^u of (2.19) for all $i \in [N]$. Then by the same arguments as in the proof above and Corollary 2.13 we get that these define upper bounds on the solution of (2.16).

Consider the function

$$R(h, p, \tilde{p}, q) = \tilde{p} - p + \tfrac{h}{2}\left[\varphi\big(p, q\big) + \varphi\big(\tilde{p}, q\big)\right].$$

Then for all $i \in [N]$ the approximations p_i^u can be computed by solving the equation $R(h, p_{i-1}^u, \tilde{p}, q) = 0$ for $\tilde{p}$. For $\tilde{p} = p_{i-1}^u$ we get

$$R(h, p_{i-1}^u, p_{i-1}^u, q) = h\,\varphi(p_{i-1}^u, q) < 0.$$

Moreover, differentiating R yields $\partial_{\tilde{p}} R(h, p, \tilde{p}, q) = 1 + \frac{h}{2}\,\partial_p\varphi(\tilde{p}, q)$ and since $\partial_p\varphi \geq 0$ this implies $\partial_{\tilde{p}} R \geq 1$. Hence, R is strictly increasing in $\tilde{p}$ and there exists a unique solution p_i^u for all $i \in [N]$. □

Remark 2.20. Since the trapezoidal rule is implicitly given by

$$R(h, p_{i-1}^u, p_i^u, q) = 0$$

for all $i \in [N]$ and this equation cannot be solved for p_i^u analytically, we have to solve the equation numerically, e.g., by Newton's method. When using Newton's method, then a new iterate p_i^n is given by

$$p_i^n = p_i^{n-1} - \frac{R(h, p_{i-1}^u, p_i^{n-1}, q)}{\partial_{\tilde{p}} R(h, p_{i-1}^u, p_i^{n-1}, q)},$$

However, note that R is strictly increasing in p_i^u as seen above and strictly concave in p_i^u if $q > 0$. Thus,

$$R(h, p_{i-1}^u, p_i^n, q) < R(h, p_{i-1}^u, p_i^{n-1}, q) + \partial_{\tilde{p}} R(h, p_{i-1}^u, p_i^{n-1}, q)\big(p_i^n - p_i^{n-1}\big) = 0$$

holds independently of p_i^{n-1}. That is, Newton's method produces lower bounds $p_i^n < p_i^u$ on the exact solution.

We point out that with a straightforward implementation of Newton's method we actually observed solutions p_N^ℓ and p_N^u of the explicit midpoint method and the trapezoidal rule with $p_N^u < p_N^\ell$! If the number of grid points $N + 1$ is big, the numerical error produced by the Newton's method can add up significantly, even when solving each step of the trapezoidal rule with Newton's method and a tolerance of 10^{-7} Pa. Therefore, we will present a variant of Newton's method in Section 6.1, which produces solutions that are greater or equal to the exact solution p_i^u for all $i \in [N]$.

Remark 2.21. Note that for Corollaries 2.18 and 2.19 to hold, it is essential that the fraction $\frac{c\,q}{A\,p}$ is bounded from above by $\nu_c \in (0, 1)$, which is strictly less than 1. Otherwise, the derivative $\partial_p \varphi(p, q)$ would not be bounded and we could not apply Corollaries 2.9 and 2.13.

Remark 2.22. Instead of applying the methods (2.18) and (2.19) in opposite direction of the flow, we can also use them to compute bounds in the direction of the flow. That is, instead of the end value problem (2.16) we can also consider the initial value problem, where we fix the pressure $p(0)$. Then the explicit midpoint method produces upper bounds and the trapezoidal rule produces lower bounds, if there are solutions p_i^ℓ and p_i^u for all grid points x_i.

However, the problem is that we cannot guarantee that for all $i \in [N]$ approximations p_i^ℓ and p_i^u with $c\,q \le \nu_c A\,p_i^\ell$ respectively $c\,q \le \nu_c A\,p_i^u$ exist. In particular, if we start with a small input pressure, then there might not exist solutions for all i. If both methods fail to produce solutions, which fulfill this bound, then we can deduce that the input pressure is too small. Otherwise, if only the trapezoidal rule fails to produce a lower bound, e.g., if we choose the input pressure such that for the analytical solution $p(L) = \frac{c\,q}{\nu_c A}$ holds, then we cannot decide whether the discretization is too coarse or the input pressure is infeasible. Therefore, we compute the methods in opposite direction of the flow.

Next, we consider the two input-output functions P^ℓ, $P^u \colon U \times \mathbb{N} \to \mathbb{R}$ defined through evaluating (2.18) and (2.19). That is,

$$P^\ell(p^0, q, N) \coloneqq p^\ell_N \qquad \text{and} \qquad P^u(p^0, q, N) \coloneqq p^u_N.$$

Moreover, we denote with $P(p^0, q) = p(0)$ the unique solution $p(x)$ of (2.16) with initial value p^0 and mass flow $q \geq 0$ evaluated at $x = 0$. We can derive the following properties for P^ℓ and P^u; see also Figure 2.1.

Lemma 2.23. *For every $\nu_c \in (0, 1)$ there exists a maximal step size $\bar{h} > 0$ such that the functions P^ℓ and P^u are nondecreasing, continuously differentiable and convex in (p, q) for all N sufficiently big with $0 < \frac{L}{N} = h \leq \bar{h}$.*

Furthermore, every solution $p(x)$ of the differential equation (2.16) with $(p^0, q) \in U$ satisfies the inequality

$$P^\ell\big(p(L), q, N\big) \;\leq\; P(p^0, q) = p(0) \;\leq\; P^u\big(p(L), q, N\big). \tag{2.20}$$

Additionally $P^\ell\big(p(L), q, N\big)$ and $P^u\big(p(L), q, N\big)$ converge to $p(0)$ for $N \to \infty$.

For instance, if $\nu_c = 0.4$, then the step size has to satisfy $0 < h \leq 4.925\frac{D}{\lambda}$.

Proof. The inequality (2.20) follows from Corollaries 2.18 and 2.19. Furthermore, under the assumptions of Corollary 2.19 the properties of P^u follow directly from Lemma 2.14.

However, we cannot apply Lemma 2.10, since the assumption of $\partial_y f$ being non-negative is not satisfied. Thus, it remains to show that there exists $\bar{h}$ such that $P^\ell \colon U \times \mathbb{N} \to \mathbb{R}$ is differentiable, nondecreasing and convex for N sufficiently big.

With $p^\ell_0 = p^\ell_0(p, q) = p$, we can write (2.18) as

$$p^\ell_i(p, q) = p^{em}\big(p^\ell_{i-1}(p, q), q, h\big) \coloneqq p^\ell_{i-1}(p, q) - h\,\varphi\big(p^\ell_{i-1}(p, q) - \tfrac{h}{2}\varphi(p^\ell_{i-1}(p, q), q), q\big)$$

for $i \in [N]$. Differentiating yields $\partial_p\, p^\ell_0(p, q) = 1$, $\partial_q\, p^\ell_0(p, q) = 0$ and

$$\begin{aligned}
\partial_p p^\ell_i(p, q) &= \partial_p p^{em}\big(p^\ell_{i-1}(p, q), q, h\big)\, \partial_p p^\ell_{i-1}(p, q), \\
\partial_q p^\ell_i(p, q) &= \partial_p p^{em}\big(p^\ell_{i-1}(p, q), q, h\big)\, \partial_q p^\ell_{i-1}(p, q) + \partial_q p^{em}\big(p^\ell_{i-1}(p, q), q, h\big)
\end{aligned}$$

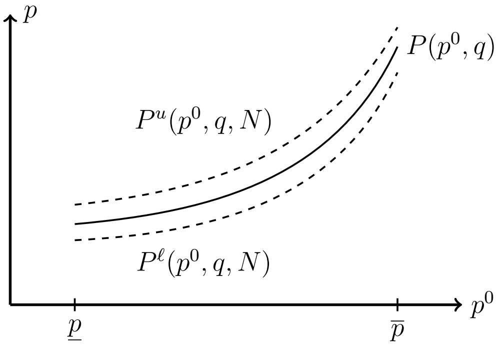


Figure 2.1. The figure depicts the properties of the input-output functions P^ℓ and P^u (dashed lines) for a fixed mass flow $q \geq 0$. The functions are nondecreasing and convex in (p^0, q). Moreover, they define a tube around the analytical solution $P(p^0, q) = p(0)$ of (2.16) (solid line). See also Lemma 2.23.

for all $i \in [N]$, where $\partial_p p^{em}$ and $\partial_q p^{em}$ denotes the partial derivative of p^{em} with respect to the first and second argument, respectively. Moreover, we have $\nabla^2 p_0^\ell(p, q) = 0$ and

$$\begin{aligned}\nabla^2 p_i^\ell(p, q) =& \nabla \begin{pmatrix} p_{i-1}^\ell(p, q) \\ q \end{pmatrix}^\top \nabla^2 p^{em}\big(p_{i-1}^\ell(p, q), q, h\big)\, \nabla \begin{pmatrix} p_{i-1}^\ell(p, q) \\ q \end{pmatrix} \\ &+ \partial_p p^{em}\big(p_{i-1}^\ell(p, q), q, h\big)\, \nabla^2 p_{i-1}^\ell(p, q)\end{aligned}$$

for all $i \in [N]$, where $\nabla^2 p^{em}$ denotes the Hessian matrix w.r.t. p and q only. Hence, we obtain by induction that $\partial_p\, p_i^\ell(p, q) \geq 0$ and $\nabla^2 p_i^\ell(p, q)$ is positive semidefinite, if $\partial_p p^{em}(p_{i-1}^\ell(p, q), q, h) \geq 0$ and $\nabla^2 p^{em}(p_i^\ell(p, q), q, h)$ is positive semidefinite. If additionally $\partial_q p^{em}(p_{i-1}^\ell(p, q), q, h) \geq 0$ holds, then also $\partial_q\, p_i^\ell(p, q) \geq 0$ follows.

Since $\varphi(p, q) \leq 0$ on U^φ, we obtain by (2.18) that $p_i^\ell(p, q) \geq p_{i-1}^\ell(p, q)$ and thus $(p_i^\ell(p, q), q) \in U^\varphi$ for $i \in [N]$. Moreover, (2.18) yields

$$\partial_p p^{em}(p, q, h) = 1 - h\, \partial_p \varphi\big(p - \tfrac{h}{2}\varphi(p, q), q\big)\big(1 - \tfrac{h}{2}\, \partial_p \varphi(p, q)\big).$$

By Lemma 2.16 we have $\partial_p \varphi \geq 0$, $\partial_{pp} \varphi \leq 0$ on U^φ and by choosing the step size h according to the assumptions of Corollary 2.18 we get $1 - \frac{h}{2}\, \partial_p \varphi(p, q) \geq 0$ and $1 - h\, \partial_p \varphi(p, q) \geq 0$ for $(p, q) \in U^\varphi$. This shows that

$$\partial_p p^{em}(p, q, h) \geq 1 - h\, \partial_p\, \varphi\big(p - \tfrac{h}{2}\varphi(p, q), q\big) \geq 1 - h\, \partial_p\, \varphi(p, q) \geq 0.$$

Moreover, one can verify that $\nabla^2 p^{em}(p,q)$ is singular and thus is positive semidefinite on U^φ if $\partial_{pp} p^{em} + \partial_{pp} p^{em} \geq 0$ on U^φ.

To show the latter, we observe that $(\partial_{pp} p^{em} + \partial_{qq} p^{em})(p,q,h)$ is a rational function in p, q and h with positive denominator on U^φ. The numerator is a polynomial which can be written in the form

$$h\left(b_0(p,q) - b_1(p,q)\,h - b_2(p,q)\,h^2 - b_3(p,q)\,h^3 - b_4(p,q)\,h^4 - b_5(p,q)\,h^5\right) =: h\,B(p,q,h)$$

where b_0 to b_5 are polynomials in p and q. These polynomials and all their derivatives w.r.t. the pressure p are positive on U^φ and since the degree of b_0 w.r.t. p is greater than the degree of the other polynomials, the derivatives of $B(p,q,h)$ w.r.t. p are positive at $p = \frac{c\,q}{\nu_c A}$, $h = 0$ and nonincreasing in h. Thus, we can determine $\bar{h}$ which satisfies the upper bound of Corollary 2.18 such that $B(p,q,h)$ and all its derivatives w.r.t. p are nonnegative for step sizes h with $0 < h \leq \bar{h}$ and $(p,q) \in U^\varphi$.

This shows that for $0 < h \leq \bar{h}$ the numerator – and thus $\partial_{pp} p^{em} + \partial_{pp} p^{em}$ – is nonnegative for all $(p,q) \in U^\varphi$. Note that then $\partial_{pp} p^{em}$ as well as $\partial_{pp} p^{em}$ are nonnegative. As already observed, this implies by induction that $\partial_p p_i^\ell(p,q) \geq 0$ and that $\nabla^2 p_i^\ell(p,q)$ is positive semidefinite, i.e., $p_i^\ell(p,q)$ is convex, for all $(p,q) \in U$.

Finally, $\partial_q p^{em}(p,0,h) = 0$ and $\partial_{qq} p^{em}(p,q,h) \geq 0$ on U^φ implies $\partial_q p^{em}(p,q,h) \geq 0$ and we deduce $\partial_q p_i^\ell(p,q) \geq 0$.

Since $\partial_p \varphi$ is bounded on U^φ and both methods have consistency order 2 they are convergent by Theorem 3.4 and Corollary 3.5 in Chapter 3 of Mattheij and Molenaar [97]; see also Remark 2.8. Hence, we deduce that $P^\ell(p(L),q,N)$ and $P^u(p(L),q,N)$ converge to $p(0)$ for $N \to \infty$. Moreover, the particular bound on the step size for $\nu_c = 0.4$ follows by determining the maximal step size such that p^{em} is convex as discussed above. □

Remark 2.24. As we already observed in Remark 2.21 it is a necessary assumption that $\frac{c\,q}{A\,p}$ is bounded from above by $\nu_c \in (0,1)$. Moreover, the suitable step sizes h and thus the discretization strongly depend on ν_c.

We have provided explicit bounds on h for $\nu_c = 0.4$ in the results above. For the gas network example `GasLib-40` which will be discussed in Section 4.6, the upper bound $h \leq 4.925\frac{D}{\lambda}$ leads to step sizes between 150 m and 570 m corresponding to 6 up to 259 grid points per pipeline. As already mentioned before, the velocity v of the gas is typically much smaller than the speed of sound c; see Section 1.1. Due to $\frac{v}{c} = \frac{c\,q}{A\,p}$, we could also use a smaller bound, e.g., $\nu_c = 0.2$. Then the upper bound on h in Lemma 2.23 would be $h \leq 29.15\frac{D}{\lambda}$. Again, for the network `GasLib-40` this corresponds to step sizes 875 m and 3.4 km.

Furthermore, also larger values ν_c are possible, but then the maximal step size decreases significantly. For example, using $\nu_c = 0.8$ leads to step sizes $h \leq 0.16\frac{D}{\lambda}$ which corresponds to 5.12 m up to 19.2 m for the network `GasLib-40`.

In this section, we have seen that the functions P^ℓ and P^u can be used to define a relaxation for constraints given by the ODE constraints (1.13). In Chapter 4 we will use these functions where we apply the spatial branch-and-bound framework which we will develop in the next chapter. To this end, we construct linear under- and overestimators for the functions P^ℓ and P^u. Since both are convex and continuously differentiable, linear underestimators for P^ℓ are given by gradient cuts. Moreover, the concave envelope of P^u over U is piecewise affine linear and it suffices to evaluate P^u at the vertices of the polytope U in order to compute the concave envelope; see, e.g., Horst and Tuy [73, Theorem IV.6].

2.4 Gas Flow in Pipelines with Height Differences

So far we have only considered pipelines without height differences. In this case $p(x)$ is concave, see Corollary 2.17, the input-output function $(p(L), q) \mapsto p(0)$ is convex and under the conditions of Lemma 2.23 the explicit midpoint method (2.18) yields a convex lower bound and the trapezoidal rule (2.19) a convex upper bound. In this section, we discuss the more general case of nonzero slope. Then $p(x)$ is not necessarily concave anymore, for example, see Figure 2.2, such that we cannot apply Corollary 2.9 to show that the explicit midpoint method defines a lower or upper bound. Instead we will distinguish three particular cases and show that the second-order Taylor method (2.12) and the trapezoidal rule (2.14) can still be used to obtain convex lower and upper bounds.

Recall that for slope $\sigma \in [-1, 1]$ the Euler equation (1.8) is given by

$$\partial_x p(x) \left(1 - \frac{c^2 q^2}{A^2 p(x)^2}\right) = -\frac{\lambda c^2}{2DA^2} \frac{q|q|}{p(x)} - \frac{g}{c^2} \sigma\, p(x), \quad x \in [0, L].$$

We still assume that $q \geq 0$ and $c\, q \leq \nu_c A\, p$ holds with $\nu_c \in (0, 1)$. Then the ODE is given by

$$\partial_x p(x) = \varphi_\sigma\big(p(x), q\big) := -\frac{p(x)}{2c^2 D} \frac{2Dg\sigma A^2 p^2(x) + \lambda c^4 q^2}{A^2 p^2(x) - c^2 q^2}, \quad x \in [0, L].$$

In the case of a nonnegative slope $\sigma \geq 0$, the right-hand side is always negative. Otherwise, if $\sigma < 0$, the right-hand side has the root $p_r(q, \sigma) := \frac{c^2 q}{A}\sqrt{\frac{-\lambda}{2Dg\sigma}}$, and $\varphi_\sigma(p, q) \geq 0$ for $p \geq p_r(q, \sigma)$ and $\varphi_\sigma(p, q) \leq 0$ for $p \leq p_r(q, \sigma)$ holds.

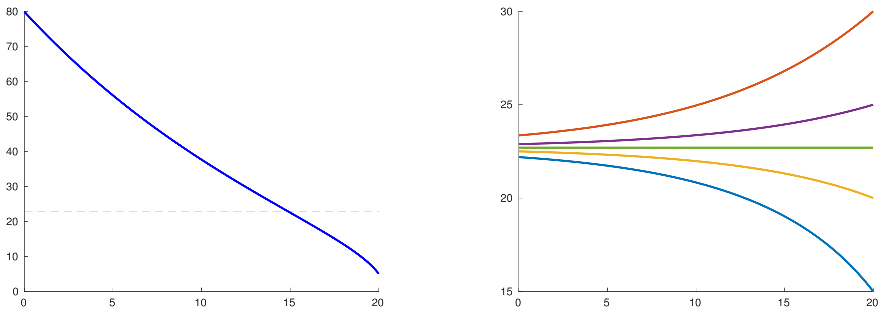


Figure 2.2. The figure shows the pressure $p(x)$ in bar along a 20 km pipe. The left figure depicts $p(x)$ for one initial value and positive slope, and the right figure shows $p(x)$ for different initial values and negative slope.

Figure 2.2 shows the change in pressure of gas flowing along a pipe with positive slope on the left and a pipe with negative slope on the right. For $\sigma > 0$ the gas has to compensate friction and gravitation, which results in an increased pressure drop in comparison with the case $\sigma = 0$. In the case $\sigma < 0$, gravitation works contrary to friction, such that there is no pressure drop or increase if the pressure $p(0)$ equals $p_r(q, \sigma)$ (see the middle line on the right of Figure 2.2). If the pressure $p(0)$ is less than the root, the pressure drop is less than in the case of $\sigma = 0$. Furthermore, if the pressure $p(0)$ is larger than the root, the pressure increases in the flow direction.

Analogously to the proof of Corollary 2.18, we consider the transformed differential equation

$$\tilde{p}(0) = p^0, \quad \partial_x \tilde{p}(x) = -\varphi_\sigma\big(\tilde{p}(x), q\big), \quad x \in [0, L] \tag{2.21}$$

with $\tilde{p}(x) = p(L - x)$ to show that we can compute lower and upper bounds on the inflow pressure $p(0) = \tilde{p}(L)$. Again, we consider an equidistant discretization with step size $h = \frac{L}{N}$. Then the application of the Taylor method yields

$$p_0^{ta} = p^0, \qquad p_i^{ta} = p_{i-1}^{ta} - h\varphi_\sigma(p_{i-1}^{ta}, q) + \frac{h^2}{2}(\partial_p \varphi_\sigma \cdot \varphi_\sigma)(p_{i-1}^{ta}, q) \qquad \forall i \in [N] \tag{2.22}$$

and application of the trapezoidal rule is given by

$$p_0^{tr} = p^0, \qquad p_i^{tr} = p_{i-1}^{tr} - \frac{h}{2}\left[\varphi_\sigma(p_{i-1}^{tr}, q) + \varphi_\sigma(p_i^{tr}, q)\right] \qquad \forall i \in [N] \tag{2.23}$$

with φ_σ instead of φ. Analogously, to the previous sections we define the input-output functions P^{ta}, $P^{tr}: U \times \mathbb{N} \to \mathbb{R}$ through evaluating (2.22) and (2.23). Note that we do not use superscript ℓ and u here, since we will see that the lower and upper bound are not always given by the same method.

To show that the second-order Taylor method and the trapezoidal rule produce convex lower and upper bounds on the solution of (2.21), we analyze the following three cases.

1. positive slope $\sigma > 0$,
2. negative slope $\sigma < 0$ with $p^0 \leq p_r(q, \sigma)$, or $\sigma = 0$,
3. negative slope $\sigma < 0$ with $p^0 \geq p_r(q, \sigma)$.

In Lemmas 2.12 and 2.14 we have seen that the second-order Taylor method and the trapezoidal rule define convex input-output functions, if $f(s, x, y)$ and $f_h^{ta}(s, h, x, y)$ are convex in (x, y). We show that these conditions are satisfied for (2.22) and (2.23) in all three cases.

Lemma 2.25. *Let $\lambda c^2 \geq -2Dg\sigma$. Then $-\varphi_\sigma(p, q)$ is convex on U^φ. Moreover, we define*

$$\varphi_h^{ta}(p, q) := -\varphi_\sigma(p, q) + \tfrac{h}{2}(\partial_p \varphi_\sigma \cdot \varphi_\sigma)(p, q).$$

Then the increment function $\varphi_h^{ta}(p, q)$ is convex on U^φ, if $\sigma > 0$ and the step size h satisfies

$$0 < h \leq \frac{D}{\lambda} \frac{(3\nu_c^2 + 1)(1 - \nu_c^2)^2}{3\nu_c^2(\nu_c^4 + 5\nu_c^2 + 2)} =: \frac{1}{b_+}, \tag{2.24}$$

or otherwise if $\sigma \leq 0$ and h satisfies

$$0 < h \leq \frac{D}{\lambda} \frac{2(3\nu_c^2 + 1)(1 - \nu_c^2)^2}{3\nu_c^2(2\nu_c^4 + 5\nu_c^2 + 1)} =: \frac{1}{b_-}. \tag{2.25}$$

For instance, if $\nu_c = 0.4$, then the step size has to satisfy $0 < h \leq 0.76\frac{D}{\lambda}$ if $\sigma > 0$ and $0 < h \leq 2.35\frac{D}{\lambda}$ if $\sigma \leq 0$.

Proof. Independently of $\sigma \in [-1, 1]$ the Hessian matrix $-\nabla^2\varphi_\sigma(p, q)$ is singular and thus positive semidefinite on U^φ if the trace $-\partial_{pp}\varphi_\sigma - \partial_{qq}\varphi_\sigma$ is nonnegative. In fact, if the inequality $\lambda c^2 \geq -2Dg\sigma$ holds, then both derivatives $\partial_{pp}\varphi_\sigma$ and $\partial_{qq}\varphi_\sigma$ are nonpositive, i.e., $-\varphi_\sigma$ is convex on U^φ.

Furthermore, also $\nabla^2\varphi_h^{ta}(p, q)$ is singular and $\partial_{pp}\varphi_h^{ta}(p, q) + \partial_{qq}\varphi_h^{ta}(p, q)$ is a rational function in (p, q, h) with positive denominator on U^φ. The numerator is a linear function in the step size h and positive for $h = 0$. Solving the numerator equals zero for h yields an upper bound on h such that $\partial_{pp}\varphi_h^{ta}(p, q) + \partial_{qq}\varphi_h^{ta}(p, q)$ is nonnegative. This upper bound is decreasing in σ and nondecreasing in p. Then evaluating this bound at $p = \frac{c\,q}{\nu_c A}$, $\sigma = 0$ and $\sigma = 1$, respectively, produces the upper bounds given in the statement. Thus, $\partial_{pp}\varphi_h^{ta}(p,q) + \partial_{qq}\varphi_h^{ta}(p,q)$ is nonnegative and $\nabla^2\varphi_h^{ta}(p,q)$ is positive semidefinite if h satisfies the upper bounds on the step size, i.e., $\varphi_h^{ta}(p,q)$ is convex on U^φ. □

Before we show that the input-output functions are not only convex, but also provide lower and upper bounds, note that further derivatives of $\tilde{p}(x)$ are given by

$$\begin{aligned} \partial_{xx}\tilde{p}(x) &= (\partial_p\varphi_\sigma \cdot \varphi_\sigma)(\tilde{p}(x), q), \\ \partial_{xxx}\tilde{p}(x) &= -\big(\partial_{pp}\varphi_\sigma(\varphi_\sigma)^2 + (\partial_p\varphi_\sigma)^2\,\varphi_\sigma\big)(\tilde{p}(x), q). \end{aligned} \tag{2.26}$$

Moreover, the derivative $-\partial_p\varphi_\sigma$ on U^φ is bounded by

$$\tfrac{1-3\nu_c^2}{c^2(1-\nu_c^2)}\, g\sigma - \tfrac{\lambda\,\nu_c^2}{2D}\,\tfrac{(\nu_c^2+1)}{(1-\nu_c^2)^2} = -\partial_p\varphi_\sigma\big(\tfrac{c\,q}{\nu_c A}, q\big) \le -\partial_p\varphi_\sigma(p,q) \le -\partial_p\varphi_\sigma(p,0) \le \tfrac{g\sigma}{c^2} \tag{2.27}$$

since $-\partial_{pp}\varphi_\sigma$ is nonnegative and $-\partial_{pq}\varphi_\sigma$ is nonpositive.

Lemma 2.26. *Let $\sigma > 0$, let b_+ be defined by* (2.24) *and define*

$$b := \frac{1-3\nu_c^2}{c^2(1-\nu_c^2)}\, g\sigma - \frac{\lambda}{2D}\,\frac{\nu_c^2(\nu_c^2+1)}{(1-\nu_c^2)^2} \quad \textit{and} \quad B := \frac{g\sigma}{c^2}.$$

Then if $h \cdot \max\{-b, B\} < 2$, the trapezoidal rule produces upper bounds on the solution of (2.21) *for $(p^0, q) \in U$ and in particular $P^{tr}\colon U \times \mathbb{N} \to \mathbb{R}$ is convex on U.*

Moreover, if $h \cdot \max\{-b, b_+\} \le 1$, the second-order Taylor method produces lower bounds and $P^{ta}\colon U \times \mathbb{N} \to \mathbb{R}$ is convex on U.

For instance, if $\nu_c = 0.4$ and $\lambda c^2 \ge 2Dg\sigma$, then P^{tr} is convex and produces upper bounds for $h \le 4\frac{D}{\lambda}$. Besides, P^{ta} is convex and produces lower bounds for step sizes $h \le 0.76\frac{D}{\lambda}$.

Proof. In the case $\sigma > 0$, we have $\varphi_\sigma(p,q) < 0$ and $\partial_{pp}\varphi_\sigma(p,q) \le 0$ and thus $\partial_{xxx}\tilde{p}(x) \ge 0$ holds, i.e., $\partial_x\tilde{p}(x)$ is convex. However, $\partial_p\varphi_\sigma(p,q)$ may change sign and thus $\tilde{p}(x)$ may change from concave to convex in x, see the left-hand side of Figure 2.2.

By inequality (2.27) we have that $-\partial_p\varphi_\sigma$ is bounded from below by b and from above by B. Additionally, $-\varphi_\sigma$ is convex on U^φ and $\partial_x\tilde{p}(x)$ is convex. Analogously to Corollary 2.19 we can show that a solution of (2.23) exists for all $i \in [N]$. Each step of the trapezoidal rule is given by

$$R(h, p_{i-1}^{tr}, p_i^{tr}, q) = p_i^{tr} - p_{i-1}^{tr} + \tfrac{h}{2}\left[\varphi_\sigma\big(p_{i-1}^{tr}, q\big) + \varphi_\sigma\big(p_i^{tr}, q\big)\right] = 0.$$

Moreover, we have $R(h, p_{i-1}^{tr}, p_{i-1}^{tr}, q) < 0$ and $R(h, p_{i-1}^{tr}, p_i^{tr}, q)$ is strictly increasing in p_i^{tr} if $h \cdot \max\{-b, B\} < 2$ holds. Hence, there exists a solution to (2.23), and by Corollary 2.13 and Lemma 2.14 the trapezoidal rule produces upper bounds and the input-output function P^{tr} is convex on U if h satisfies $h \cdot \max\{-b, B\} < 2$.

Furthermore, by Lemma 2.25 we get that $\varphi_h^{ta}(p, q)$ is convex for $h \leq b_+^{-1}$. Besides $(\partial_{pp}\varphi_\sigma \cdot \varphi_\sigma + (\partial_p \varphi_\sigma)^2)$ is nonnegative on U^φ, that is, Corollary 2.11 holds with $b' = 0$ and shows together with Lemma 2.12 that the Taylor method produces a convex lower bound if h satisfies $h \cdot \max\{-b, b_+\} \leq 1$.

The particular bounds for $\nu_c = 0.4$ follow by evaluating b, B and b_+ at $\nu_c = 0.4$ and underestimating the bounds by using $\lambda c^2 \geq 2Dg\sigma$. $\square$

Lemma 2.27. *Let $\sigma < 0$ and $p^0 \leq p_r(q, \sigma)$, or let $\sigma = 0$. Moreover, let the inequality $\lambda c^2 \geq -2Dg\sigma$ be satisfied and b, b_- as defined in Lemmas 2.25 and 2.26. Then if inequality $h \cdot \max\{-b, 0\} \leq 2$ is true and a solution of (2.23) with $p_i^{tr} \leq p_r(q, \sigma)$ for all $i \in [N]$ exists, the trapezoidal rule produces upper bounds on the solution of (2.21) for $(p^0, q) \in U$ and in particular $P^{tr}: U \times \mathbb{N} \to \mathbb{R}$ is convex on U.*

Moreover, if $h \cdot \max\{-b, b_-\} \leq 1$, the second-order Taylor method produces lower bounds and $P^{ta}: U \times \mathbb{N} \to \mathbb{R}$ is convex on U.

For instance, if $\nu_c = 0.4$, then P^{tr} defines a convex upper bound for $h \leq \frac{D}{\lambda}\frac{441}{24}$ and P^{ta} defines a convex lower bound for step sizes $h \leq 2.35\frac{D}{\lambda}$.

Proof. This case is analogous to the analysis of $\sigma > 0$. Here, we have that $\varphi_\sigma(p, q) \leq 0$, $\partial_p \varphi_\sigma(p, q) \geq 0$ and $\partial_{pp}\varphi_\sigma(p, q) \leq 0$ hold on U^φ with $p \leq p_r(q, \sigma)$, therefore $\partial_{xx}\tilde{p}(x) \leq 0$ and $\partial_{xxx}\tilde{p}(x) \geq 0$ also hold, i.e., $\tilde{p}(x)$ is concave and $\partial_x \tilde{p}(x)$ is convex. Moreover, the signs of φ_σ and its derivatives yield:

$$(\partial_{pp}\varphi_\sigma \cdot \varphi_\sigma + (\partial_p \varphi_\sigma)^2)(p, q) \geq 0.$$

Again, by Lemma 2.25 $-\varphi_\sigma$ is convex. Further, the partial derivative $-\partial_p \varphi_\sigma$ is bounded by $b \leq -\varphi_\sigma \leq 0$. Thus, by Corollary 2.13 and Lemma 2.14 the trapezoidal rule produces upper bounds and the input-output function P^{tr} is convex on U if h satisfies $h \cdot \max\{-b, 0\} \leq 2$.

Similar to before Lemma 2.25 yields that $\varphi_h^{ta}(p, q)$ is convex for $h \leq b_-^{-1}$. Besides $(\partial_{pp}\varphi_\sigma \cdot \varphi_\sigma + (\partial_p \varphi_\sigma)^2)$ is nonnegative on U^φ, that is, Corollary 2.11 holds again with $b' = 0$ and shows together with Lemma 2.12 that the Taylor method produces a convex lower bound if h satisfies $h \cdot \max\{-b, b_-\} \leq 1$. Furthermore, the particular bounds for $\nu_c = 0.4$ follow by evaluating b and b_- at $\nu_c = 0.4$. $\square$

Lemma 2.28. *Let $\sigma < 0$ and $p^0 \geq p_r(q,\sigma)$. Moreover, let $\lambda c^2 \geq -6Dg\sigma$ and suppose that $p_r(p,q) \geq \frac{c\,q}{\nu_c A}$. Then if*

$$0 < h \leq \frac{\lambda c^2 + 2Dg\sigma}{-\lambda g\sigma}$$

and a solution of (2.23) with $p_i^{tr} \geq p_r(q,\sigma)$ for all $i \in [N]$ exists, the trapezoidal rule produces lower bounds on the solution of (2.21) for $(p^0,q) \in U$ and the input-output function $P^{tr} \colon U \times \mathbb{N} \to \mathbb{R}$ is convex on U.

Moreover, if h satisfies

$$0 < h \leq \frac{\lambda c^2 + 2Dg\sigma}{-3\lambda g\sigma},$$

the second-order Taylor method produces upper bounds and $P^{ta} \colon U \times \mathbb{N} \to \mathbb{R}$ is convex on U.

Otherwise, if $p_r(p,q) < \frac{c\,q}{\nu_c A}$, then the upper bounds on the step sizes are given by the conditions of Lemma 2.27, i.e., the upper bounds are given by $h \cdot \max\{-b,\, 0\} \leq 2$ for the trapezoidal rule and $h \cdot \max\{-b,\, b_-\} \leq 1$ for the second-order Taylor method with b and b_- as defined in Lemmas 2.25 and 2.26.

Proof. We only consider the case $p_r(p,q) \geq \frac{c\,q}{\nu_c A}$. The other case follows analogously. In this case, we have $\varphi_\sigma(p,q) \geq 0$, $\partial_p \varphi_\sigma(p,q) \geq 0$ and $\partial_{pp}\varphi_\sigma(p,q) \leq 0$. Hence, $\tilde{p}(x)$ is convex and $-\partial_p\varphi_\sigma$ is bounded by

$$0 \geq -\partial_p\varphi_\sigma\big(p,q\big) \geq -\partial_p\varphi_\sigma\big(p_r(q,\sigma),q\big) = \frac{2\lambda g\sigma}{2Dg\sigma + \lambda c^2}.$$

However, unlike before the signs of φ_σ and its derivatives do not immediately imply that $\partial_x\tilde{p}(x)$ is convex or concave. Nevertheless, we can show that in this case derovative $\partial_x\tilde{p}(x)$ is concave. In fact, since $\partial_{xxx}\tilde{p}(x) = -(\partial_{pp}\varphi_\sigma(\varphi_\sigma)^2 + (\partial_p\varphi_\sigma)^2\,\varphi_\sigma)(\tilde{p}(x),q)$ and $\varphi_\sigma(p,q) \geq 0$ holds, it suffices to show that

$$d(p,q) \coloneqq \big(\partial_{pp}\varphi_\sigma \cdot \varphi_\sigma + (\partial_p\varphi_\sigma)^2\big)(p,q) \geq 0$$

is true. To this end, we observe that $d(p,q)$ is a rational function with positive denominator on U^φ. The numerator of $d(p,q)$ is a polynomial of degree 8 in p and q. Moreover, we can show that the numerator is nonnegative on U^φ if $\lambda c^2 \geq -6Dg\sigma$ is satisfied.

Hence, $\partial_x\tilde{p}(x)$ is concave and the trapezoidal rule yields by Corollary 2.13 lower bounds if the step size h satisfies

$$0 < h \leq \frac{\lambda c^2 + 2Dg\sigma}{-\lambda g\sigma}$$

and by Lemma 2.14 the input-output function is convex. Moreover, since $d(p,q) \geq 0$ is true, Corollary 2.11 holds with $b' = 0$ and thus the second-order Taylor methods produces upper bounds if the step size h satisfies

$$0 < h \leq \frac{\lambda c^2 + 2Dg\sigma}{-3\lambda g\sigma}.$$

To see that P^{ta} is convex too, note that the increment function $\varphi_h^{ta}(p,q)$ is convex for step size $h \leq \frac{1}{b_-}$; see Lemma 2.25. However, to determine b_- we have used that the pressure is bounded below by $p \geq \frac{c\,q}{\nu_c A}$. But by assumption $p_r(q,\sigma) \geq \frac{c\,q}{\nu_c A}$ holds. Hence, using $p \geq p_r(q,\sigma)$ instead yields that $\varphi_h^{ta}(p,q)$ is convex under the bound above. Thus, by Lemma 2.12 the second-order Taylor produces a convex input-output function P^{ta}, too. $\square$

Remark 2.29. In Corollary 2.9 and Lemma 2.26 we could show that there exist solutions to the implicit trapezoidal rule. However, in Lemmas 2.27 and 2.28 we assumed that solutions p_i^{tr} with $p_i^{tr} \leq p_r(q,\sigma)$, respectively, $p_i^{tr} \geq p_r(q,\sigma)$ for all $i \in [N]$ exist. The reason for that is the following. Each step of the implicit trapezoidal rule is given by

$$R(h, p_{i-1}^{tr}, p_i^{tr}, q) = p_i^{tr} - p_{i-1}^{tr} + \tfrac{h}{2}\left[\varphi_\sigma\big(p_{i-1}^{tr}, q\big) + \varphi_\sigma\big(p_i^{tr}, q\big)\right] = 0.$$

However, if $\sigma < 0$ and $p_{i-1}^{tr} - \frac{h}{2}\,\varphi_\sigma(p_{i-1}^{tr}, q)$ is greater, respectively, less than $p_r(q,\sigma)$, there exist no p_i^{tr} with the properties above. Nevertheless, in both cases $p_r(q,\sigma)$ is an upper, respectively, lower bound on the analytical solution $p(0)$.

Note that, in case 3, i.e., $\sigma < 0$ and $p^0 \geq p_r(q,\sigma)$, we could avoid this problem by considering $p(0) = p^0$ and computing bounds on $p(L)$. But in case 2 we cannot avoid this problem, since $p(L)$ has to satisfy the inequality $\frac{c\,q}{A\,p(L)} \leq \nu_c$.

Lemmas 2.26 to 2.28 show that we can compute lower and upper bounds on the solution of the stationary isothermal Euler equations even in the case with height differences. We remark that the assumption of $\lambda c^2 \geq 6Dg|\sigma|$ poses only a restriction on very small friction coefficients λ, since we typically have $c^2 \gg 6Dg|\sigma|$.

In Chapter 4, we develop a spatial branch-and-bound algorithm to solve optimization problems for stationary gas transport which is based on the relaxations produced by the explicit midpoint method, second-order Taylor method and the trapezoidal rule. Therefore, we distinguish between $\sigma = 0$ and $\sigma \neq 0$ again. At first, we develop the algorithm for the case $\sigma = 0$ and then discuss how to adapt the algorithm to cope with slope $\sigma \neq 0$; see Section 4.4 and Section 4.5.

2.5 Outlook

The results in this chapter leave a lot of open questions and possibilities for future research. First of all, the conditions stated in Corollaries 2.9, 2.11 and 2.13 are rather strict, since they require the derivative of the solution of the scalar differential equation to be convex or concave and in Corollary 2.9 also the solution itself. Moreover, there is the obvious question if these results can be generalized to systems of ODEs.

Concerning the latter question, we can present the following analogue of Lemma 2.3 which provides a sufficient condition for one-step methods to produce lower or upper bounds on the solution of an ODE system.

Lemma 2.30. *Consider a method of the form* (2.3) *for a system of ODEs with continuously differentiable* $f\colon \mathbb{R} \times \mathbb{R}^k \times \mathbb{R}^n \to \mathbb{R}^n$. *Let the local discretization error of the method be nonnegative, i.e., the inequality*

$$y(s+h) - y(s) - h\, f_h\big(s, h, x, y(s), y(s+h)\big) \geq 0$$

holds for all $s \in [0, S]$ *and* $h \geq 0$ *with* $s + h \leq S$. *Define the mean value derivatives*

$$\hat{D}_y f_h(s, h, x, y, \tilde{y}, z, \tilde{z}) := \int_0^1 \partial_y f_h(s, h, x, y + \mu(z - y), \tilde{y} + \mu(\tilde{z} - \tilde{y}))\, \mathrm{d}\mu,$$
$$\hat{D}_{\tilde{y}} f_h(s, h, x, y, \tilde{y}, z, \tilde{z}) := \int_0^1 \partial_{\tilde{y}} f_h(s, h, x, y + \mu(z - y), \tilde{y} + \mu(\tilde{z} - \tilde{y}))\, \mathrm{d}\mu.$$

Suppose there are $h_{max} > 0$ *and* $d_{max} > 0$ *such that*

$$\Big(I - h\, \hat{D}_{\tilde{y}} f_h(s, h, x, y(s), y(s+h), z, \tilde{z})\Big)^{-1} \Big(I + h\, \hat{D}_y f_h(s, h, x, y(s), y(s+h), z, \tilde{z})\Big)$$

has nonnegative entries for all $0 < h \leq h_{max}$, $s \in [0, S-h]$, $\|z - y(s)\| \leq d_{max}$, *and* $\|\tilde{z} - y(s+h)\| \leq d_{max}$. *Then for all* $0 < h \leq h_{max}$ *such that the solution of the method* (2.3) *satisfies* $\|y_i - y(s_i)\| \leq d_{max}$, $i \in [N]$, *one has* $y_i \leq y(s_i)$ *for all* $i \in [N]$.

Otherwise, if the local discretization error is nonpositive, then we obtain $y_i \geq y(s_i)$ *for all* $i \in [N]$, *under the same assumptions.*

Note that this result has already been given in [56] and can be proven in a similar way to Lemma 2.3.

In Section 2.3, we have seen that we can successfully apply Corollaries 2.9 and 2.13 to the stationary isothermal Euler equations without height differences even though their requirements are rather strict. However, it seems that this particular differential equation has several favorable properties, i.e., concavity of the solution and its derivatives. Nevertheless, the more complicated case of nonzero slope in Section 2.4 shows that it is still possible to derive lower and upper bounds by analyzing the properties of the right-hand side of the ODE and the properties of the solution if these requirements are not fulfilled for all initial values and parameter.

Furthermore, for the context of bound propagation this example yields another idea. Consider the case depicted on the left-hand side of Figure 2.2, i.e., positive slope σ and positive mass flow q. In this case, there exists p^* with $\partial_p \varphi_\sigma(p^*, q) = 0$ and $x^* \in [0, L]$ with $p(x^*) = p^*$. Moreover, the solution $p(x)$ is convex for $x \leq x^*$ and concave for $x \geq x^*$.

Given a discretization $0 = x_N < \ldots < x_1 < x_0 = L$, we can analogously to Corollary 2.18 show that the explicit midpoint method produces lower bounds p_i^ℓ on $p(x_i)$ for all $i \in [N]$ with $p_i^\ell \leq p^*$. Suppose that there is an index $j \in [N-1]$ with $p_j^\ell < p^*$ but $p_{j+1}^\ell \geq p^*$. Then we can (numerically) compute a new step size $h_{j+1}^* \leq h_{j+1}$ such that

$$p_{j+1}^{\ell,*} = p^{em}(p_j^\ell, q, h_{j+1}^*) = p^* \leq p(x_j - h_{j+1}^*)$$

holds. If we additionally have a method which provides lower bounds in the remaining part of the pipeline, for example, the second-order Taylor method, we can concatenate these two methods and thus compute a lower bound on $p(0)$.

Next, we consider a general autonomous ODE. Then the second derivative of the solution is $\partial_{ss} y(s) = \partial_y f\big(x, y(s)\big) f\big(x, y(s)\big)$. If bounds on the possible solutions are known, we can check whether the right-hand side and its derivatives change their signs in this interval. Note that such bounds are often known, e.g., bounds on the amount of substances in a chemical reaction. Therefore, the set of possible solution values can be partitioned in such a way that the solution is either convex or concave on each part. Constructing under- and overestimators for every part provides lower and upper bounds on the whole solution.

We point out that even if it may not be possible to prove convexity of the input-output functions for the concatenation of different methods, the lower and upper bounds can be used to check δ-feasibility as we will do in the next chapter and for bound propagation. Moreover, bound propagation can be used to construct a *finitely consistent* bounding operation such that spatial branch-and-bound terminates finitely; see Horst and Tuy [73, Theorem IV.1].

CHAPTER 3

Spatial Branch-and-Bound for ODE Constrained Problems

In this chapter, we develop a spatial branch-and-bound algorithm to globally solve mixed-integer nonlinear optimization problems with parameter-dependent ordinary differential equation constraints of the form

$$\begin{aligned} \min \quad & C(x, y^0, y^S, z) \\ \text{s.t.} \quad & G(x, y^0, y^S, z) \le 0, \\ & \partial_s y(s) = f\big(s, x, y(s)\big), \qquad s \in [0, S], \\ & y^0 = y(0),\ y^S = y(S), \\ & x \in X,\ y^0 \in Y^0,\ y^S \in Y^S,\ z \in Z, \end{aligned} \tag{3.1}$$

where $X \subset \mathbb{R}^k$ and Y^0, $Y^S \subset \mathbb{R}^n$ are polytopes and $Z \subset \mathbb{Z}^m$ is bounded. The objective function $C\colon X \times Y^0 \times Y^S \times Z \to \mathbb{R}$ as well as the constraints given by the function $G\colon X \times Y^0 \times Y^S \times Z \to \mathbb{R}^l$ can be nonlinear, but we assume that C, G, and the function $f\colon \mathbb{R} \times \mathbb{R}^k \times \mathbb{R}^n \to \mathbb{R}^n$ are continuously differentiable. The variables $y(s)$ are functions that solve the ODEs specified by $\partial_s y(s) = f(s, x, y(s))$ for $s \in [0, S]$. Note that we assume that $y(s)$ can be the solution of a single ODE system, of a collection of (independent) ODE systems or even n scalar differential equations. Moreover, continuous variables x, y^0, y^S and integer variables z are present.

The particular structure of (3.1) is motivated by the application of stationary gas networks which will be the topic of the subsequent chapter. This class of optimization problems has clear connections to mixed-integer optimal control problems and global optimization of dynamical systems, however, the distinguishing feature of (3.1) is that the solution(s) $y(s)$ of the differential equations only need to be known at a finite number of positions, namely 0 and S. The objective function and further constraints of (3.1) only depend on the corresponding values y^0, y^S and the parameters x, but not on the ODE solution $y(s)$ at some intermediate point $s \in (0, S)$. Note that for notational simplicity, we assume that the ODEs are defined on the same interval $[0, S]$; this can be assured by reparametrization.

A natural approach for optimization problems like (3.1) is the *first-discretize-then-optimize* approach. That is, after discretizing the differential equations in (3.1) the resulting mixed-integer nonlinear problem is solved. However, in this approach one has only an a priori guarantee on the distance of a feasible solution of the discretized problem to a feasible solution of the original problem. Furthermore, one has no guarantee that the original problem is infeasible if the discretized problem is infeasible, since discretization does not yield a relaxation of the feasible set. Moreover, using fine discretizations to compute accurate solutions leads to eventually huge MINLPs which can then be hard to solve due to their size. Instead, we define relaxations of the original problem. Hence, infeasibility of the relaxations certifies infeasibility of the original problem.

The main ideas of this chapter are based on the assumption that functions F^ℓ and $F^u\colon X \times Y^0 \times \mathbb{N}^n \to \mathbb{R}^n$ exist which are under- and overestimators for the input-output function of the differential equation in (3.1), that is, the function

$$F\colon X \times Y^0 \to \mathbb{R}^n, \quad \big(x, y^0\big) \mapsto y(S)$$

which maps the parameters x and the initial values $y^0 = y(0)$ to the solution $y(S)$ at the second boundary. Moreover, we assume that F^ℓ and F^u converge to F for $N \to \infty$. This assumption is clearly motivated by the results of the previous chapter, where we have seen that in particular cases one-step methods for scalar ODEs can be used to define such functions F^ℓ and F^u. Then with these functions we define a relaxation of (3.1) by replacing the differential equations with

$$F^\ell(x, y^0, N) \;\le\; y^S \;\le\; F^u(x, y^0, N).$$

We start this chapter with a short literature review. Afterwards, in Section 3.2 we present an equivalent reformulation of problem (3.1). We then introduce the required assumptions on the functions F^ℓ and F^u mentioned above which are used to relax the reformulation and then show how (ε, δ)-optimal solutions of the reformulation and the relaxation are related. In Section 3.3 we show that spatial branch-and-bound can be applied to the relaxation under some standard assumptions and discuss two approaches how to compute (ε, δ)-optimal solutions for the reformulation of (3.1). These approaches are similar to first-discretize-then-optimize approaches in the sense that the parameter N of the relaxation either has to be chosen up front or iteratively refined to achieve a desired accuracy. Finally, in Section 3.4 we develop an algorithm which incorporates adaptively changing the parameter N in a single spatial branch-and-bound tree.

We remark that the results of this chapter have been presented in [56] which is joint work with Marc E. Pfetsch and Stefan Ulbrich. This chapter has been supplemented with a discussion of how to extend Algorithm 3.3 such that the parameter N can be chosen more freely during the course of the algorithm and how to define a consistent feasibility notion in Section 3.4.

3.1 Literature Review

In this chapter, we develop a method to globally solve a class of optimization problems of the form (3.1). The general approach is to use branch-and-bound to handle the integer variables z and spatial branching for handling nonlinearities. Both approaches are standard in mixed-integer nonlinear programming, see, for example, the books by Horst and Tuy [73], by Lee and Leyffer [87], and by Locatelli and Schoen [93] or the overview articles of Floudas and Gounaris [38], Hemmecke et al. [66], Belotti et al. [10], and more recently by Kılınç and Sahinidis [79]. The basic idea of branch-and-bound methods is to recursively divide the feasible set into smaller parts, i.e., nodes, (branching) and solve convex relaxations of the original problem on these parts to derive lower bounds on the optimal solution value. If the solution of the convex relaxation is a feasible solution of the original problem, the solution also defines an upper bound on the optimal solution value (bounding). Based on lower and upper bounds on the optimal solution value, a node can be fathomed if the lower bound on this part is greater than the currently best known upper bound. In this process, spatial branching refers to the technique in which the domain of a continuous variable is split into (usually two) nonempty parts and thereby creating (two) new child nodes in the so-called branch-and-bound tree. Since the bounds on the variable are tighter in each child node, the hope is that this can be used by other solver components to further tighten bounds. This process produces tighter relaxations in the child nodes and results in a convergent solution algorithm under appropriate conditions.

For nonlinear optimization problems the convex relaxations are usually based on convex underestimators and concave overestimators of the constraints and it seems that the first time the convex envelope, i.e., the largest convex underestimator, has appeared in the literature is in Kleibohm [81]. Some further early articles which use convex envelopes or underestimators for global optimization are the ones by Falk [33], Falk and Soland [35], Falk and Hoffman [34] and Horst [72]. Since then some well-known techniques have been established to derive convex, concave or even linear under- and overestimators for large classes of functions, e.g., McCormick inequalities [99], outer-approximation by Duran and Grossmann [30], the reformulation-linearization technique by Sherali and coworkers [137, 138, 139], or the αBB method by Adjiman and coworkers [3, 4]. Moreover, the McCormick inequalities have been generalized by Scott et al. [136, 145]. Other articles often deal with underestimators for particular functions or with specific properties; e.g., see Rikun [113], Liberti and Pantelides [89], Tardella [147], Meyer and Floudas [101], or Tawarmalani et al. [149].

The class of optimization problems (3.1) has clear connections to mixed-integer optimal control problems with ordinary differential equation as well as partial differential equation constraints and global optimization of dynamical systems. Since the focus of this thesis lies on ODE constrained problems, we only review articles that deal with such problems here. For a starting point on optimization with PDE constraints, see, for example, Hinze et al. [70] or the articles mentioned in Section 1.4. Approaches to solve ODE constrained problems are often based on the first-discretize-then-optimize approach and a partial list of articles which use this approach for ODE constrained problems is as follows. Čižniar et al. [24] proposed a method that uses a time discretization and a polynomial basis to represent the solutions between the time points. Sager et al. [120] developed a convexification method to handle specific discrete decisions over time that switch the right-hand sides of the differential equations (e.g., gear shifting) and show how to efficiently compute feasible solutions; if the corresponding continuous relaxation is solved to global optimality, then such solutions converge to a global optimal solution while refining the discretization. Extending this approach, Sager and coworkers [119, 121], and Jung et al. [76] developed a solution algorithm for the so-called combinatorial integral approximation problem. Zeile and coworkers [122, 156, 157] further investigated combinatorial integral approximation decompositions and constraints to avoid unrealistic frequent switching of integer decision. Furthermore, Kirches et al. [80] extended the partial outer convexification approach to include additional constraints on the solutions of the ODEs and the integer decisions. Bock et al. [15] considered problems with implicit and explicit switches. Based on first-discretize-then-optimize, they provided a reformulation as a nonlinear program with vanishing constraints, which they solve numerically. Using the αBB approach, Diedam and Sager [28] developed a method to globally solve the nonlinear programs arising from a multiple-shooting discretization for optimal control problems without integer decisions. Wilhelm et al. [155] presented a spatial branch-and-bound approach for global optimization of stiff dynamical systems. After discretizing the differential equations they apply the convexification methods of Stuber et al. [145] and solve the resulting problem by spatial branch-and-bound. All these first-discretize-then-optimize approaches use a fixed discretization, i.e., the solutions only provide an approximation of the solutions of the ODEs with respect to an a priori fixed accuracy. Thus, the discretization error is ignored or it is implicitly assumed that the discretization is refined if an a posteriori accuracy check fails. Hante and Schmidt [60] provide sufficient conditions such that the optimal value converges if the discretization is iteratively refined.

Global optimization approaches for problems with ODE constraints have been proposed in the following articles. Esposito and Floudas [31] developed an approach based on a fixed discretization of the control and αBB relaxations of the solution operator which maps the control to the ODE solution. Papamichail and Adjiman [109, 110] considered parametric ODEs and construct approximations via the αBB approach. They proposed a spatial branch-and-bound algorithm based on solving nonlinear programs (NLPs) in which ODEs have to be solved within the solution of the NLPs. Lin and Stadtherr [91] used a time-discretization technique to enclose ODE solutions in a branch-and-bound algorithm; see also Section 2.1. Moreover, continuous-time enclosure techniques, see Section 2.1, have been used or studied for the use in branch-and-bound methods by Chachuat et al. [21, 22], Barton and Singer [141], Scott et al. [135], and Scott and Barton [133]. One further deterministic global optimization approach has been developed by Bajaj and Hasan [6]. In [64] Hasan presented a method to construct edge-concave underestimators for functions over rectangles which is similar to the αBB method. Then these underestimators can be linearly underestimated to define convex relaxations for spatial branch-and-bound. This method has been extended by Bajaj and Hasan [6] to construct edge-concave underestimators for functions depending on parametric ODE solutions.

The two articles by Gugat et al. [53] and Schmidt et al. [128] present global optimization approaches based on decomposition which are related to our approach. Since they apply their methods to a stationary gas transport problem similar to ours, these articles are discussed in more detail in Section 4.1.

The spatial branch-and-bound algorithm that we present in this chapter is distinct from the approaches mentioned above in the following way. We adaptively refine the discretization, i.e., the parameter N, which is not done in the approaches based on first-discreteize-then-optimize. Moreover, we exploit the particular assumption that the ODE solutions only needs to be known at a finite number of points to derive lower and upper bounds on the ODE solutions which is different from the general-purpose approximations for ODEs and the convexifications mentioned above.

3.2 Relaxation Hierarchy

In this section, we introduce the basic assumptions needed for our approach, an equivalent reformulation of optimization problem (3.1) and relaxations of this reformulation which will be the actual problems solved in the spatial branch-and-bound algorithm we develop. Moreover, we show how we can use the relaxations to compute so-called (ε, δ)-optimal solutions for the equivalent reformulation of (3.1).

We begin with a natural assumption on the existence of solutions of the differential equations which we assume to hold throughout this chapter.

Assumption 1. The initial value problem

$$y(0) = y^0, \quad \partial_s y(s) = f\big(s, x, y(s)\big), \quad s \in [0, S] \tag{3.2}$$

is uniquely solvable for all $x \in X$ and $y^0 \in Y^0$.

This assumption is guaranteed, for example, if f is Lipschitz continuous w.r.t. y for all parameters x. We then denote by

$$F \colon X \times Y^0 \to \mathbb{R}^n, \quad \big(x, y^0\big) \mapsto y(S),$$

the solution operator/input-output function of the initial value problem (3.2) which maps the parameters $x \in X$ and initial values $y^0 \in Y^0$ to the corresponding unique solution $y(S)$ at the boundary. Replacing the ODE constraints in (3.1) by

$$y^S - F(x, y^0) = 0$$

yields the equivalent problem

$$\begin{aligned} \min \quad & C(x, y^0, y^S, z) \\ \text{s.t.} \quad & G(x, y^0, y^S, z) \leq 0, \\ & y^S - F\big(x, y^0\big) = 0, \\ & x \in X,\ y^0 \in Y^0,\ y^S \in Y^S,\ z \in Z. \end{aligned} \tag{3.3}$$

The problems (3.1) and (3.3) are equivalent in the sense that every feasible or optimal solution $(x, y^0, y^S, y(s), z)$ of the former problem defines a feasible or optimal solution (x, y^0, y^S, z) of the latter problem. Moreover, for every feasible or optimal solution (x, y^0, y^S, z) of problem (3.3) there exists by construction a solution $y(s)$ of the ODE system (3.2) such that $(x, y^0, y^S, y(s), z)$ is a feasible or optimal solution of (3.1).

Note that by Assumption 1 and the assumption that f is continuously differentiable we can derive that F depends continuously on x and y^0; e.g., see Hartman [61].

Thus, since X, Y^0, Y^S are polytopes, Z is bounded and C, G are continuous as well, the problem has an optimal solution if the feasible set is nonempty. Furthermore, if there is an algebraic formula for F, then we could in principle use a black-box spatial branch-and-bound solver to solve (3.3). Our approach is motivated by the assumption that either such a formula is not known or the formula is (too) hard to evaluate. For example, Gugat et al. [54] have shown that the stationary isothermal Euler equation (1.8) with $\sigma = 0$ admits an analytical solution, however, the solution has to be evaluated numerically, e.g., by Newton's method, and thus is not suited for standard MINLP techniques.

Our idea is to construct under- and overestimators of $F(x, y^0)$ by the right choice of suitable numerical methods. Motivated by the results in the previous chapter we make the following assumption.

Assumption 2. There exist continuous functions $F^\ell \colon X \times Y^0 \times \mathbb{N}^n \to \mathbb{R}^n$ and $F^u \colon X \times Y^0 \times \mathbb{N}^n \to \mathbb{R}^n$ which fulfill the inequality

$$F^\ell\big(x, y^0, N\big) \;\le\; F\big(x, y^0\big) \;\le\; F^u\big(x, y^0, N\big)$$

for all $x \in X$, $y^0 \in Y^0$ and $N \ge N_0 \in \mathbb{N}^n$. Furthermore, we assume that the functions F_i^ℓ and F_i^u converge uniformly to F_i for all $i \in [n] = \{1, \dots, n\}$ if $N_i \to \infty$.

For the example of stationary gas transport the functions F^ℓ and F^u can for every pipeline be defined by P^ℓ and P^u given by the evaluation of the explicit midpoint method and the trapezoidal rule. Thereby, N denotes the number of grid points in an equidistant discretization and since the methods are convergent P^ℓ and P^u converge to the solution P for $N \to \infty$; see Lemma 2.23.

We then relax the constraint $y^S = F(x, y^0)$ of the problem (3.3) by means of the functions F^ℓ and F^u. In this way we derive the problem

$$\begin{aligned} \min\;\; & C(x, y^0, y^S, z) \\ \text{s.t.}\;\; & G(x, y^0, y^S, z) \le 0, \\ & F^\ell\big(x, y^0, N\big) \le y^S \le F^u\big(x, y^0, N\big), \\ & x \in X,\; y^0 \in Y^0,\; y^S \in Y^S,\; z \in Z. \end{aligned} \tag{3.4}$$

For $N \ge N_0$ this is a relaxation of (3.3), since every feasible point of (3.3) is feasible for the new constraint

$$F^\ell\big(x, y^0, N\big) \le y^S \le F^u\big(x, y^0, N\big),$$

and the objective function is the same. Note that this constraint and thus (3.4) depends on $N \in \mathbb{N}^n$, yet we assume that N is not an optimization variable. Again, the optimal value of this problem is bounded from below if the feasible set is nonempty, because X, Y^0, Y^S are polytopes, Z is bounded, and C, G, F^ℓ and F^u are continuous.

One of the main steps in (spatial) branch-and-bound algorithms is to compute lower bounds on the optimal value; see, for example, Horst and Tuy [73], Locatelli and Schoen [93], or Kılınç and Sahinidis [79]. Therefore, usually convex relaxations of the original problem are solved, that is, a so-called *convex underestimator* of the objective function is minimized on a convexification of the feasible set. Thereby, a convex underestimator is defined as follows.

Definition 3.1. *Let $W \subseteq \mathbb{R}^d$ and consider a function $g\colon W \to \mathbb{R}$. We say that a function $\check{g}\colon \operatorname{conv}(W) \to \mathbb{R}$ is a* convex underestimator *of g if $\check{g}$ is convex and $\check{g} \leq g$ holds on W. Analogously, a concave function $\hat{g}\colon \operatorname{conv}(W) \to \mathbb{R}$ with $\hat{g} \geq g$ on W is called a* concave overestimator *of g.*

Moreover, we call $\check{g}\colon \operatorname{conv}(W) \to \mathbb{R}^l$ a convex underestimator or concave overestimator of a vector-valued function $g\colon W \to \mathbb{R}^l$ if $\check{g}_i$ is a convex underestimator, respectively, concave overestimator of g_i for all $i \in [l]$.

Assumption 3. We assume that we can construct convex underestimators $\check{G}$ and $\check{C}$ of G and C and in particular a convex underestimator $\check{F}^\ell$ and a concave overestimator $\hat{F}^u$ of F^ℓ and F^u for all $N \geq N_0 \in \mathbb{N}^n$ on all subsets $\tilde{X} \times \tilde{Y}^0 \times \tilde{Y}^S \times \tilde{Z}$ of $X \times Y^0 \times Y^S \times Z$ which result from $X \times Y^0 \times Y^S \times Z$, e.g., via branching.

To derive convex, concave or even linear under- and overestimators there exist well-known techniques in the literature, e.g., McCormick inequalities [99], outer-approximation by Duran and Grossmann [30], the reformulation-linearization technique by Sherali and coworkers [137, 138, 139], or the αBB method by Adjiman and coworkers [3, 4]. Other articles often deal with the construction of underestimators based specific properties or with underestimators for particular functions; e.g., see Meyer and Floudas [101] or Tardella [147] who deal with edge-concave functions or Liberti and Pantelides [89] who construct underestimators for monomials of odd degrees.

Usually these techniques are automatically applied by state-of-the-art solvers for mixed-integer nonlinear programming, for example, ANTIGONE [102], BARON [123], COUENNE [11], or SCIP [40] which are all based on spatial branch-and-bound. However, they cannot be used as black-box solvers for problems containing the functions F^ℓ and F^u defined by one-step methods for initial value problems as discussed in the previous chapter. Nevertheless, it is still possible to derive under- and overestimators for these functions. If F^ℓ and F^u are defined by explicit one-step methods, then under- and overestimators can be derived by iteratively under- and overestimating each step. Otherwise, if they are based on implicit methods, then under- and overestimators can be derived by generalized McCormick relaxations for implicit functions, see Scott et al. [136, 145]. Furthermore, if the input-output functions are already convex, then linear underestimators for F^ℓ are given by gradient cuts and the concave envelope, i.e., the tightest concave overestimator, is a piecewise linear function whose construction only requires the evaluation of F^ℓ at the vertices of $X \times Y^0$; see Horst and Tuy [73, Theorem IV.6] or Section 4.3 where we construct such under- and overestimators for the example of stationary gas transport.

Based on the assumption above, we obtain the following convex relaxation of (3.4):

$$\begin{aligned}
\min \quad & \alpha \\
\text{s.t.} \quad & \check{C}(x, y^0, y^S, z) - \alpha \leq 0, \\
& \check{G}(x, y^0, y^S, z) \leq 0, \\
& \check{F}^\ell\big(x, y^0, N\big) \leq y^S \leq \hat{F}^u\big(x, y^0, N\big), \\
& x \in X,\ y^0 \in Y^0,\ y^S \in Y^S,\ z \in \operatorname{conv}(Z).
\end{aligned} \tag{3.5}$$

Since (3.5) is a relaxation of optimization problem (3.3) which is a reformulation of (3.1), this problem can be seen as a convex relaxation of our original problem (3.1). Moreover, we assume that this problem can actually be solved by MINLP solvers like the ones mentioned above. For example, SCIP solves relaxations of the MINLP where $\check{C}$, $\check{G}$, $\check{F}^\ell$ and $\hat{F}^u$ are all given by linear functions.

Remark 3.2. Another possibility to derive a relaxation of (3.1) is to use the continuous-time enclosure techniques by Singer and Barton [140], Scott and Barton [134], Scott et al. [135, 136] and Harwood and Barton [62, 63]; see Section 2.1. These articles consider parametric ODEs of the form

$$\partial_s y(s,x) = f\big(s, x, y(s,x)\big), \qquad y(0,x) = y_0(x),$$

where the initial value is given by a continuous function $y_0(x)$ depending on the parameter x. The authors derive auxiliary ODE systems

$$\begin{aligned}
\partial_s c(s,x) &= u\big(s, x, c(s,x), C(s,x)\big), \qquad & c(0,x) &= c_0(x), \\
\partial_s C(s,x) &= o\big(s, x, c(s,x), C(s,x)\big), \qquad & C(0,x) &= C_0(x),
\end{aligned}$$

such that the solutions satisfy $c(s,x) \leq y(s,x) \leq C(s,x)$. Furthermore, the solutions c and C are convex, respectively, concave w.r.t. the parameter x. Moreover, these systems can be used in a spatial branch-and-bound algorithm; for example, see Chachuat et al. [21, 22].

These relaxations may also be used to define the functions F^ℓ and F^u. However, then the ODE system still either has to be solved with arbitrary precision, i.e., we loose the desired adaptivity of our approach, or additionally has to be discretized by bound preserving methods as introduced in the previous chapter.

Spatial branch-and-bound will enable us to compute so-called (ε, δ)-*optimal solutions* of the relaxation (3.4). In the following we show how these are related to (ε, δ)-optimal solutions of (3.3). This will then give rise to a first idea how to compute such solutions. To this end, for a vector $y \in \mathbb{R}^n$ we denote with $(y)_+$ the vector of the componentwise maxima of y_i and 0 and we define (ε, δ)-optimal solutions as follows.

Definition 3.3. *Let $\varepsilon > 0$ and $\delta > 0$. We say that $(x, y^0, y^S, z) \in X \times Y^0 \times Y^S \times Z$ is a δ-*feasible solution *of (3.3) if the following condition holds:*

$$\max\left\{ \left\| \left(G(x, y^0, y^S, z)\right)_+ \right\|_\infty,\ \left\| y^S - F(x, y^0) \right\|_\infty \right\} \le \delta.$$

*Analogously, we call $(x, y^0, y^S, z) \in X \times Y^0 \times Y^S \times Z$ a δ-*feasible solution *of (3.4) if instead*

$$\max\Big\{ \left\| \left(G\left(x, y^0, y^S, z\right)\right)_+ \right\|_\infty, \left\| \left(F^\ell\left(x, y^0, N\right) - y^S\right)_+ \right\|_\infty, \\ \left\| \left(y^S - F^u\left(x, y^0, N\right)\right)_+ \right\|_\infty \Big\} \le \delta.$$

*holds. Furthermore, we call $(x, y^0, y^S, z) \in X \times Y^0 \times Y^S \times Z$ an (ε, δ)-*optimal solution *of (3.3) or (3.4) if it is δ-feasible and the objective function satisfies*

$$C(x, y^0, y^S, z) \le C^* + \varepsilon,$$

where $C^* > -\infty$ is the optimal value of (3.3) or (3.4), or $C^* = \infty$ if their respective feasible set is empty.

Note that this definition is consistent with the definition in the literature, e.g., in Locatelli and Schoen [93]. Since our goal is to find (ε, δ)-optimal solutions of (3.3) by approximatively solving (3.4), we now show how their respective (ε, δ)-optimal solutions are related.

Lemma 3.4. *Let $\delta_1,\ \delta_2 > 0$. Let $(x, y^0, y^S, z) \in X \times Y^0 \times Y^S \times Z$ be an (ε, δ_1)-optimal solution of (3.4) for some $N \ge N_0 \in \mathbb{N}^n$. Additionally, suppose that the* *condition*

$$\left\|F^u(x,y^0,N) - F^\ell(x,y^0,N)\right\|_\infty \le \delta_2 \tag{3.6}$$

is satisfied. Then (x, y^0, y^S, z) *is an* (ε,δ)*-optimal solution of* (3.3) *for all* $\delta \ge \delta_1 + \delta_2$.

Proof. First, we prove that (x, y^0, y^S, z) is δ-feasible of problem (3.3). Because of the δ_1-feasibility for (3.4), we know that $\left\|\left(G(x,y^0,y^S,z)\right)_+\right\|_\infty \le \delta_1 \le \delta$ as well as

$$\left(y^S - F^u(x,y^0,N)\right)_+ + \left(F^\ell(x,y^0,N) - y^S\right)_+ \le \delta_1$$

is true. Here, we used that $F^u \ge F^\ell$ holds and for all $i \in [n]$ either $y_i^S - F_i^u(x,y^0,N)$ or $F_i^\ell(x,y^0,N) - y_i^S$ can be positive. Thus, we can derive

$$\begin{aligned}
\left|y_i^S - F_i(x,y^0)\right| &= \Big(y_i^S - F_i(x,y^0)\Big)_+ + \Big(F_i(x,y^0) - y_i^S\Big)_+ \\
&\le \Big(y_i^S - F_i^u(x,y^0,N)\Big)_+ + \Big(F_i^u(x,y^0,N) - F_i(x,y^0)\Big)_+ \\
&\qquad + \Big(F_i(x,y^0) - F_i^\ell(x,y^0,N)\Big)_+ + \Big(F_i^\ell(x,y^0,N) - y_i^S\Big)_+ \\
&= \Big(y_i^S - F_i^u(x,y^0,N)\Big)_+ + F_i^u(x,y^0,N) \\
&\qquad - F_i^\ell(x,y^0,N) + \Big(F_i^\ell(x,y^0,N) - y_i^S\Big)_+ \le \delta_1 + \delta_2 \le \delta.
\end{aligned}$$

That is, (x, y^0, y^S, z) is a δ-feasible solution of (3.3).

Let $(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z})$ be an optimal solution of (3.3). Since problem (3.4) is a relaxation of (3.3), the solution $(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z})$ is a feasible solution of (3.4). Hence, the feasible set is nonempty and there exists an optimal solution $(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$ of (3.4) with $C(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z}) \le C(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z})$. Since (x, y^0, y^S, z) is an (ε, δ_1)-optimal solution of the relaxation (3.4), we can derive

$$C(x,y^0,y^S,z) \le C(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z}) + \varepsilon \le C(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z}) + \varepsilon,$$

that is, (x, y^0, y^S, z) is an (ε, δ)-optimal solution of (3.3). Otherwise, if (3.3) is infeasible, the condition $C(x, y^0, y^S, z) \le C^* + \varepsilon = \infty$ is obviously satisfied. □

Lemma 3.4 immediately shows how to generate an (ε, δ)-optimal solution of (3.3). If the under- and overestimators fulfill certain technical conditions, we can compute an (ε, δ_1)-optimal solution of (3.4) by spatial branch-and-bound. Then if the solution satisfies (3.6), the solution is an (ε, δ)-optimal solution of (3.3).

3.3 Basic Spatial Branch-and-Bound Approach

In this section, we will see that spatial branch-and-bound can be used to approximatively solve problem (3.4) for fixed $N \geq N_0 \in \mathbb{N}^n$ under mild assumptions on the under- and overestimators. That is, spatial branch-and-bound applied to (3.4) produces (ε, δ)-optimal solutions or returns that the problem is infeasible in finite time. Thus, we can use spatial branch-and-bound to compute $(\varepsilon, \delta_1 + \delta_2)$-optimal solutions of (3.3) in the following two ways. Since F^ℓ and F^u converge to F by Assumption 2, one possibility is to choose N sufficiently big such that condition (3.6) holds for all $(x, y^0) \in X \times Y^0$. Then solving (3.4) with tolerances $\delta_1 > 0$ and $\varepsilon > 0$ yields an $(\varepsilon, \delta_1 + \delta_2)$-optimal solution of problem (3.3). The second possibility is to start with some $N \geq N_0 \in \mathbb{N}^n$, solve (3.4) and check if condition (3.6) is satisfied in the solution. If the condition does not hold, we can increase N and solve (3.4) again. Repeating this process until the solution finally satisfies (3.6) yields an $(\varepsilon, \delta_1 + \delta_2)$-optimal solution of (3.3) again, since by Assumption 2 the functions F^ℓ and F^u converge to F.

Applying the basic principle of spatial branch-and-bound to problem (3.4) yields Algorithm 3.1. We remark that we assume that in practice further techniques to enhance the performance of spatial branch-and-bound are used, however, we only consider the basic principle to show that the ideas above actually work. For example, see Puranik and Sahinidis [112] for a survey of domain reduction techniques, Belotti et al. [11] for branching and bound tightening techniques, or Section 6.2 for the bound tightening we implemented for the example of stationary gas transport.

We initialize Algorithm 3.1 with feasibility tolerance $\delta > 0$, tolerance $\varepsilon > 0$, and $N \geq N_0 \in \mathbb{N}^n$ sufficiently big, such that Assumptions 2 and 3 are satisfied. Note that the lower bound N_0 can for example be given by the necessary condition on the step sizes in Lemma 2.23. Furthermore, we start with the initial upper bound $\mathcal{U} = \infty$ and initialize the branch-and-bound tree with root node $X \times Y^0 \times Y^S \times Z$. In every node $\tilde{X} \times \tilde{Y}^0 \times \tilde{Y}^S \times \tilde{Z}$ of the branch-and-bound tree, we first update or construct the convex relaxation of (3.4) which is then solved. If the relaxation is infeasible on node $\tilde{X} \times \tilde{Y}^0 \times \tilde{Y}^S \times \tilde{Z}$, we can cut off the node. Otherwise, we check if the solution is a δ-feasible solution of (3.4) and update the upper bound $\mathcal{U}$ if the current solution provides a better upper bound; see Lines 9 and 10. Afterwards, if $\tilde{\alpha} < \mathcal{U} - \varepsilon$, i.e., the solution of the relaxation does not exclude the existence of a solution with an optimal value which is at least ε less than $\mathcal{U}$, we perform branching. Thereto, we can choose either an integer variable z_i whose solution $\tilde{z}_i$ is not integer, a variable x_i, y_i^0, or y_i^S appearing in a constraint which is δ-violated in the current solution, or a variable x_i, y_i^0, or y_i^S appearing in the objective function C if the underestimator $\check{C}$ is not accurate enough, that is, $C(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z}) - \check{C}(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z}) > \varepsilon$. We point out that the choice of the branching variable is not only crucial for the performance of

Algorithm 3.1 Spatial branch-and-bound for (3.4)

Input: Problem (3.4), $N \geq N_0 \in \mathbb{N}^n$, $\delta > 0$ and $\varepsilon > 0$.

Output: (ε, δ)-optimal solution $(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z})$ of (3.4) or "infeasible".

Upper bound $\mathcal{U} \leftarrow \infty$
List of active nodes $\mathcal{L} \leftarrow \{X \times Y^0 \times Y^S \times Z\}$
While $\mathcal{L} \neq \varnothing$ **do**
  choose a node $\tilde{X} \times \tilde{Y}^0 \times \tilde{Y}^S \times \tilde{Z} \in \mathcal{L}$ and set $\mathcal{L} \leftarrow \mathcal{L} \setminus \{\tilde{X} \times \tilde{Y}^0 \times \tilde{Y}^S \times \tilde{Z}\}$.
  Build and solve the convex relaxation (3.5) w.r.t. $\tilde{X} \times \tilde{Y}^0 \times \tilde{Y}^S \times \tilde{Z}$.
  If (3.5) is feasible **then**
    let $(\tilde{\alpha}, \tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$ be an optimal solution of (3.5).
    If $\tilde{z} \in \mathbb{Z}^m$ **then**
      If $(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$ is δ-feasible for (3.4) and $C(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z}) < \mathcal{U}$ **then**
        set $\mathcal{U} \leftarrow C(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$ and $(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z}) \leftarrow (\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$.
    If $\tilde{\alpha} < \mathcal{U} - \varepsilon$ **then**
      choose a branching variable according to one of the following cases:
        • An integer variable z_i with $\tilde{z}_i \notin \mathbb{Z}$.
        • A variable x_i, y_i^0 or y_i^S in a δ-violated constraint $G(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z}) \leq 0$, $F^\ell(\tilde{x}, \tilde{y}^0, N) \leq \tilde{y}^S$, or $\tilde{y}^S \leq F^u(\tilde{x}, \tilde{y}^0, N)$.
        • A variable x_i, y_i^0 or y_i^S in the objective if $C(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z}) < \tilde{\alpha} - \varepsilon$.
      Branch w.r.t. the chosen variable and add nodes to $\mathcal{L}$.
If $\mathcal{U} < \infty$ **then**
  return (ε, δ)-optimal solution $(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z})$
else
  return "infeasible".

spatial branch-and-bound, but also has to satisfy conditions necessary for the convergence of spatial branch-and-bound; see Theorem 3.5 and, for example, Achterberg et al. [2]. Finally, at the end of the algorithm we either return the best found solution or that the problem is infeasible.

To show that Algorithm 3.1 terminates finitely, we have to require the following conditions. We remark that these are standard assumptions for (spatial) branch-and-bound methods. Suppose the algorithm produces through branching an infinite nested sequence of nodes

$$\mathcal{F}_k := X_k \times Y_k^0 \times Y_k^S \times Z_k$$

with $\mathcal{F}_{k+1} \subseteq \mathcal{F}_k$ for all $k \in \mathbb{N}_0$. Then the branching rules have to satisfy the condition

$$\lim_{k\to\infty} \operatorname{diam}(\mathcal{F}_k) = 0, \tag{3.7}$$

where $\operatorname{diam}(W) := \max_{w,w'\in W} \|w - w'\|_2$ is the diameter of a bounded set $W \subset \mathbb{R}^d$. Moreover, Assumption 3 has to hold true, i.e., for every node $\mathcal{F}_k$ we need to be able to construct convex underestimators $\check{C}$, $\check{G}$, $\check{F}^\ell$ and the concave overestimator $\hat{F}^u$. In the following, we denote the dependency on $\mathcal{F}_k$ by index k, e.g., $\check{C}_k$ is a convex underestimator of C on node $\mathcal{F}_k$. Furthermore, if (3.7) holds, then for $k \to \infty$ the under- and overestimators have to satisfy

$$\begin{aligned}\max_{(x,y^0,y^S,z)\in\mathcal{F}_k} \Big\{ &\big\|G(x,y^0,y^S,z) - \check{G}_k(x,y^0,y^S,z)\big\|_\infty, \\ &\big\|F^\ell(x,y^0,N) - \check{F}_k^\ell(x,y^0,N)\big\|_\infty, \big\|\hat{F}_k^u(x,y^0,N) - F^u(x,y^0,N)\big\|_\infty, \\ &\big|C(x,y^0,y^S,z) - \check{C}_k(x,y^0,y^S,z)\big| \Big\} \to 0.\end{aligned} \tag{3.8}$$

That is, the under- and overestimators become arbitrarily close to the approximated functions over small sets. The conditions (3.7) and (3.8) are sometimes called the property of *exhaustiveness* and the property of *exactness in the limit*.

If Assumptions 1, 2 and 3, and conditions (3.7) and (3.8) hold, then Algorithm 3.1 terminates finitely, which follows directly from Theorem 5.26 in Locatelli and Schoen [93]. Thereby we assume for simplicity that the relaxations can be evaluated exactly, i.e., without rounding errors, otherwise a further approximation error would have to be handled.

Theorem 3.5. *Let $\varepsilon > 0$, $\delta > 0$ and $N \geq N_0 \in \mathbb{N}^n$. Suppose that Assumptions 1, 2 and 3, and conditions* (3.7) *and* (3.8) *are satisfied. Then Algorithm 3.1 terminates after a finite number of iterations and either returns an (ε, δ)-optimal solution of* (3.4) *or that problem* (3.4) *is infeasible.*

Note that there can exist (ε, δ)-optimal solutions of (3.3) even if (3.4) is infeasible. In this case, both results are possible. It can happen that Algorithm 3.1 finds an (ε, δ)-optimal solution or that δ-feasible solutions of (3.4) are infeasible for (3.5) and the algorithm returns “infeasible”. This is due to the fact that under- and overestimators are usually tight at some points and thus cut off δ-feasible solutions. For example, the McCormick estimators for the product of two variables over a square are exact in the corners.

Theorem 3.5 shows that both ideas for the computation of (ε, δ)-optimal solutions of (3.3) sketched in the beginning can be applied separately. That is, either by choosing N sufficiently big and then solving (3.4) with Algorithm 3.1 or by repeatedly solving the relaxation and increasing N we can derive the following result.

Corollary 3.6. *Let $\varepsilon > 0$ and feasibility tolerances δ_1, $\delta_2 > 0$. Suppose that Assumptions 1, 2 and 3 are satisfied, and conditions* (3.7) *as well as* (3.8) *hold for $N \geq N_0 \in \mathbb{N}^n$. Then we can compute an $(\varepsilon, \delta_1 + \delta_2)$-optimal solution of* (3.3) *in finite time, or establish the infeasibility of* (3.3).

Proof. On the one hand, by Assumption 2 we know that F^ℓ and F^u converge uniformly to F for $N_i \to \infty$ for all $i \in [n]$. Therefore, we can choose N sufficiently big such that

$$\left\| F^u\big(x, y^0, N\big) - F^\ell\big(x, y^0, N\big) \right\|_\infty \leq \delta_2$$

holds for all $(x, y^0) \in X \times Y^0$. Then by Theorem 3.5 the spatial branch-and-bound algorithm with parameters $\delta_1 > 0$, $\varepsilon > 0$ and N returns an (ε, δ_1)-optimal solution of (3.4) or that it is infeasible in finite time.

Since (3.4) is a relaxation of (3.3), there is no feasible solution of (3.3) if the Algorithm 3.1 returns “infeasible”. Otherwise, the algorithm returns an (ε, δ_1)-optimal solution of (3.4) and Lemma 3.4 states that this solution is an $(\varepsilon, \delta_1 + \delta_2)$-optimal solution of (3.3).

On the other hand, we can run Algorithm 3.1 with parameters $\delta_1 > 0$, $\varepsilon > 0$ and an initial $N^1 \geq N_0$. Then either the algorithm proves that problem (3.3) is infeasible or returns an (ε, δ_1)-optimal solution of (3.4) for N^1. If the algorithm returns an optimal solution $(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z})$, we can check if this solution satisfies (3.6). If not, then we can choose a vector $N^2 \geq N^1$ with $N_i^2 > N_i^1$ for all $i \in [n]$ with

$$\left| F_i^u\big(\bar{x}, \bar{y}^0, N_i^1\big) - F_i^\ell\big(\bar{x}, \bar{y}^0, N_i^1\big) \right| > \delta_2$$

and run Algorithm 3.1 again with N^2 instead of N^1. Afterwards, we repeat this process until a solution satisfies (3.6) or the relaxation is infeasible.

Since F^ℓ and F^u converge uniformly to F by Assumption 2, each N_i has to be increased only a finite number of times until condition (3.6) holds. This shows, that repeatedly solving (3.4) with increased N either stops with an $(\varepsilon, \delta_1 + \delta_2)$-optimal solution of (3.3) or with an infeasible problem (3.4) for some N^k. □

3.4 Adaptive Spatial Branch-and-Bound

Both approaches to compute (ε, δ)-optimal solutions of (3.3) described in the previous sections are similar to first-discretize-then-optimize approaches in the following sense. Either we use large values for N or we have to repeatedly apply the spatial branch-and-bound Algorithm 3.1 to achieve the desired accuracy. In the first case, we possibly choose N much larger than it has to be. This can lead to an increased computational effort, for example, if N corresponds to the number of grid points of a discretization. In the second case, we possibly perform a whole branch-and-bound process several times in vain. To circumvent these problems we propose an algorithm which incorporates changing N in a single spatial branch-and-bound algorithm.

Therefore, consider Line 5 of Algorithm 3.1. Instead of "simply" constructing the convex relaxation and then performing the δ-feasibility check for problem (3.4), we replace Line 5 with the following adaptive procedure, see Algorithm 3.2.

In a node $\tilde{X} \times \tilde{Y}^0 \times \tilde{Y}^S \times \tilde{Z}$ of the branch-and-bound tree we start with constructing the underestimators $\check{C}$ and $\check{G}$ on the current node and then run Algorithm 3.2 with feasibility tolerances δ_1, δ_2 and the current (global) parameter N. Again, we suppose that this can be done by standard methods such as McCormick inequalities [99] or the α-BB method of Adjiman and coworkers [3, 4]; see also Section 3.2. In Algorithm 3.2 we first choose an initial relaxation of inequality

$$F^\ell\big(x, y^0, N\big) \le y^S \le F^u\big(x, y^0, N\big). \tag{3.9}$$

For example, we can relax this inequality by using $\check{F}^\ell$ and $\hat{F}^u$ from the parent node (if we are not in the root node) or we can completely relax this constraint and just use the variable bounds given by $\tilde{X} \times \tilde{Y}^0 \times \tilde{Y}^S$. Then we solve the convex relaxation (3.5) on the current node. If the relaxation is infeasible, so is the corresponding original problem and we return "infeasible" to the branch-and-bound process. Otherwise, let $(\tilde{\alpha}, \tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$ be an optimal solution of the relaxation. For this solution we check if condition (3.6) holds in $(\tilde{x}, \tilde{y}^0)$ and possibly increase N until it does. Note that at this point we do not need to solve the relaxation again, since we do not update the relaxation when increasing N. Afterwards, if $(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$ is a δ_1-feasible solution of constraint (3.9), we stop the algorithm and return the current solution to the

Algorithm 3.2 Adaptive convex relaxation

Input: Node of the branch-and-bound tree $\tilde{X} \times \tilde{Y}^0 \times \tilde{Y}^S \times \tilde{Z}$, tolerances δ_1, $\delta_2 > 0$, parameter $N \in \mathbb{N}^n$, and convex underestimators $\check{C}$, $\check{G}$.

Output: A δ_1-feasible solution of (3.9) satisfying (3.6), "infeasible" or "branch".

Choose relaxation of (3.9), e.g., $\tilde{X} \times \tilde{Y}^0 \times \tilde{Y}^S$ or relaxation of parent node.
For $k = 1, 2, \ldots$ **do**
  Solve the convex relaxation (3.5) on node $\tilde{X} \times \tilde{Y}^0 \times \tilde{Y}^S \times \tilde{Z}$.
  If the relaxation is feasible **then**
    let $(\tilde{\alpha}, \tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$ be the solution of (3.5).
    For all $i \in [n]$ **do**
      While $|F_i^u(\tilde{x}, \tilde{y}^0, N_i) - F_i^\ell(\tilde{x}, \tilde{y}^0, N_i)| > \delta_2$ **do**
        increase N_i.
    If $(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$ is δ_1-feasible for (3.9) **then**
      return solution $(\tilde{\alpha}, \tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$.
    Choose the "most violated" constraint i, i.e.,

$$i \in \operatorname*{arg\,max}_{j \in [n]} \max \left\{ F_j^\ell\big(\tilde{x}, \tilde{y}^0, N_j\big) - \tilde{y}_j^S,\ \tilde{y}_j^S - F_j^u\big(\tilde{x}, \tilde{y}^0, N_j\big) \right\}.$$

    If $\tilde{y}_i^S > F_i^u\big(\tilde{x}, \tilde{y}^0, N_i\big) + \delta_1$ **then**
      • either "improve the overestimator" or
      • suggest branching w.r.t. constraint $y_i^S \leq F_i^u\big(x, y^0, N_i\big)$ and **return** "branch",
    else if $\tilde{y}_i^S < F_i^\ell\big(\tilde{x}, \tilde{y}^0, N_i\big) - \delta_1$ **then**
      • either "improve the underestimator" or
      • suggest branching w.r.t. constraint $F_i^\ell\big(x, y^0, N_i\big) \leq y_i^S$ and **return** "branch".
  else
    return "infeasible".

branch-and-bound process. Otherwise, if the solution is not δ_1-feasible, we choose the "most violated constraint" $i \in [n]$ of (3.9); see Line 11. For this constraint either

$$\tilde{y}_i^S > F_i^u(\tilde{x}, \tilde{y}^0, N_i) + \delta_1 \quad \text{or} \quad \tilde{y}_i^S < F_i^\ell(\tilde{x}, \tilde{y}^0, N_i) - \delta_1$$

holds. Subsequently, in Lines 13 and 16 we try to improve the overestimator $\hat{F}_i^u$ or the underestimator $\check{F}_i^\ell$ and thereby cut off the current solution. If we can successfully cut off the solution, then we solve the convex relaxation again and repeat the process. Otherwise, if this is not possible, we stop with the current solution and suggest to perform branching w.r.t. to this constraint to resolve the δ_1-infeasibility.

Note that in Lines 13 and 16 we can just require to improve the over- and underestimator, because the construction of $\check{F}_i^\ell$ and $\hat{F}_i^u$ depends on the particular problem. For example, if F_i^ℓ is convex or F_i^u is concave, then we can add a gradient cut to cut off the current solution. Another possibility is to add an estimator dynamically, instead of adding all inequalities at once, i.e., if an under- or overestimator consists of multiple inequalities, we can add all inequalities at once or only one inequality which separates the current solution. Again, this is, for example, possible in outer-approximation if F_i^ℓ is convex or F_i^u is concave; see Duran and Grossmann [30]. Furthermore, we point out that the relaxation might already contain an underestimator of F_i^ℓ or overestimator of F_i^u, but that it does not cut off the current solution since it has been constructed before N_i was increased.

Incorporating Algorithm 3.2 into the spatial branch-and-bound algorithm results in Algorithm 3.3. The main change is of course that N need not be constant anymore. Instead we use N as a global parameter for all nodes. That is, once N is increased in Algorithm 3.2, we use the increased parameter N in all nodes of the branch-and-bound tree which are processed afterwards. Moreover, Note that a δ_1-feasible solution of (3.4) for N might not be a δ_1-feasible solution of (3.4) for $N' \geq N$, but is still a $(\delta_1 + \delta_2)$-feasible solution of (3.3) if it fulfills condition (3.6) for N. Therefore, Algorithm 3.3 solves problem (3.3) and not (3.4). Another big difference is that we do not have to (re-)construct $\check{F}^\ell$ and $\hat{F}^u$ in every node, but instead refine them only if needed. Except for this, the algorithm is almost the same as Algorithm 3.1.

Note that Line 10 contains a hidden integrality check for $\tilde{z}$, since by Definition 3.3 the solution of the relaxation is δ_1-feasible for (3.4) if and only if $\tilde{z}$ is integral. Furthermore, if Algorithm 3.2 cannot resolve infeasibility by improving an under- or overestimator and returns "branch," we can still choose to first perform branching with respect to some integral variable or due to another violated constraint in Lines 17 and following.

The crucial point for proving that Algorithm 3.3 terminates finitely, is that Algorithm 3.2 terminates after a finite number of iterations. As this cannot be proven

Algorithm 3.3 Adaptive spatial branch-and-bound for (3.3)

Input: Problem (3.3), $N^0 \geq N_0 \in \mathbb{N}^n$, tolerances δ_1, $\delta_2 > 0$ and $\varepsilon > 0$.

Output: $(\varepsilon, \delta_1 + \delta_2)$-optimal solution $(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z})$ or "infeasible".

Upper bound $\mathcal{U} \leftarrow \infty$
List of active nodes $\mathcal{L} \leftarrow \big\{X \times Y^0 \times Y^S \times Z\big\}$
While $\mathcal{L} \neq \varnothing$ **do**
  choose a node $\tilde{X} \times \tilde{Y}^0 \times \tilde{Y}^S \times \tilde{Z} \in \mathcal{L}$ and set $\mathcal{L} \leftarrow \mathcal{L} \setminus \{\tilde{X} \times \tilde{Y}^0 \times \tilde{Y}^S \times \tilde{Z}\}$.
  Construct underestimators $\check{C}$ and $\check{G}$.
  Run Algorithm 3.2.
  If Algorithm 3.2 stops with a solution or "branch" **then**
    let $(\tilde{\alpha}, \tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$ be the last solution found in Algorithm 3.2.
    If the solution is δ_1-feasible for (3.9) **then**
      If $(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$ is δ_1-feasible for (3.4) and $C(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z}) < \mathcal{U}$ holds **then**
        set $\mathcal{U} \leftarrow C(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$ and $(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z}) \leftarrow (\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z})$.
    If $\tilde{\alpha} < \mathcal{U} - \varepsilon$ **then**
      choose a branching variable according to one of the following cases:
        - An integer variable z_i with $\tilde{z}_i \notin \mathbb{Z}$.
        - A variable x_i, y_i^0 or y_i^S in a δ-violated constraint $G(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z}) \leq 0$.
        - A variable x_i, y_i^0 or y_i^S in the objective if $C(\tilde{x}, \tilde{y}^0, \tilde{y}^S, \tilde{z}) < \tilde{\alpha} - \varepsilon$.
        - A variable x_j, y_j^0 or y_i^S in the "most violated" constraint
$$F_i^\ell\big(\tilde{x}, \tilde{y}^0, N\big) \leq \tilde{y}_i^S \quad \text{or} \quad \tilde{y}_i^S \leq F_i^u\big(\tilde{x}, \tilde{y}^0, N\big)$$
          if Algorithm 3.2 suggested to "branch".
      Branch w.r.t. the chosen variable and add nodes to $\mathcal{L}$.
If $\mathcal{U} < \infty$ **then**
  return $(\varepsilon, \delta_1 + \delta_2)$-optimal solution $(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z})$
else
  return "infeasible".

in general, but only for a given construction method of $\check{F}^\ell$ and $\hat{F}^u$, we require the following assumption.

Assumption 4. If Algorithm 3.2 keeps N fixed, then it terminates after finitely many iterations.

Note that we do not suppose that the algorithm stops with a δ_1-feasible solution, it only has to stop with either a δ_1-feasible solution, "infeasible" or "branch." The next Lemma shows that this assumption is enough to ensure that Algorithm 3.2 terminates after finitely many iterations.

Lemma 3.7. *If Assumptions 1, 2, 3 and 4 hold, then Algorithm 3.2 terminates finitely.*

Proof. Assume that the algorithm does not terminate. Then it produces a sequence of points which are feasible solutions of the convex relaxation but not δ_1-feasible for constraint (3.9). We denote by $K \subset \mathbb{N}$ the iterations where N is increased.

Since by Assumption 2 the functions F^ℓ and F^u converge uniformly to F w.r.t. N and $\tilde{X} \times \tilde{Y}^0$ is bounded, there exists $N^0 \in \mathbb{N}^n$ such that

$$\left\| F^u\big(x, y^0, N\big) - F^\ell\big(x, y^0, N\big) \right\|_\infty \leq \delta_2$$

is satisfied for all $(x, y^0) \in \tilde{X} \times \tilde{Y}^0$ and all $N \geq N^0$. That is, each N_i can only be increased a finite number of times until $N_i \geq N_i^0$ holds. Hence, K is either empty or a finite set. Thus N is fixed either from the beginning of the algorithm or after the last iteration $k \in K$. Then due to Assumption 4 the algorithm stops after another finite number of iterations. $\square$

Using this lemma we will show in Section 4.3 that Algorithm 3.2 applied to the example of gas transport, where F^ℓ and F^u are given by Lemma 2.23, terminates finitely.

In the following we state sufficient conditions such that Algorithm 3.3 terminates finitely. Again, we consider an infinite nested sequence of nodes

$$\mathcal{F}_k = X_k \times Y_k^0 \times Y_k^S \times Z_k$$

with $\mathcal{F}_{k+1} \subseteq \mathcal{F}_k$ produced by Algorithm 3.3 and corresponding vectors N^k produced by Algorithm 3.2 for all $k \geq 0$. The branching rules still have to satisfy the property of exhaustiveness (3.7), i.e.,

$$\lim_{k \to \infty} \operatorname{diam}(\mathcal{F}_k) = 0.$$

Since Algorithm 3.2 only improves the estimators if (3.9) is δ_1-violated in the current solution of the relaxation, it might happen that an estimator $\check{F}^\ell$ or $\hat{F}^u$ does not change although

$$\max_{(x,y^0,y^S,z)\in\mathcal{F}_k} \Big\{ \big\| F^\ell(x,y^0,N^k) - \check{F}^\ell_k(x,y^0,N^k) \big\|_\infty, \\ \big\| \hat{F}^u_k(x,y^0,N^k) - F^u(x,y^0,N^k) \big\|_\infty \Big\} > \delta_1$$

holds true. Thus, (3.8) cannot hold either. Instead, if (3.7) holds, we require that only the convex underestimators of C and G satisfy (3.8), i.e., for $k \to \infty$ we assume

$$\max_{(x,y^0,y^S,z)\in\mathcal{F}_k} \Big\{ \big\| G(x,y^0,y^S,z) - \check{G}_k(x,y^0,y^S,z) \big\|_\infty, \\ \big| C(x,y^0,y^S,z) - \check{C}_k(x,y^0,y^S,z) \big| \Big\} \to 0. \tag{3.10}$$

We replace the assumption on $\check{F}^\ell$ and $\hat{F}^u$ with the condition that at some point Algorithm 3.2 does not return "branch" anymore. That is, δ_1-infeasible solutions of (3.9) observed during Algorithm 3.2 can be cut off by improving $\check{F}^\ell$ or $\hat{F}^u$. Therefore, for each node $\mathcal{F}_k$ denote the last solution found in Algorithm 3.2 by $(\tilde{\alpha}^k, \tilde{x}^k, \tilde{y}^{0,k}, \tilde{y}^{S,k}, \tilde{z}^k)$. We then assume that for every sequence of nodes, which satisfies (3.7), there exists an iteration $k_0 \in \mathbb{N}$ such that the condition

$$\max\Big\{ \big\| \big(F^\ell(\tilde{x}^k, \tilde{y}^{0,k}, N^k) - \tilde{y}^{S,k}\big)_+ \big\|_\infty,\ \big\| \big(\tilde{y}^{S,k} - F^u(\tilde{x}^k, \tilde{y}^{0,k}, N^k)\big)_+ \big\|_\infty \Big\} \le \delta_1 \tag{3.11}$$

is satisfied for all $k \ge k_0$. Under these conditions we can now show that Algorithm 3.3 terminates finitely.

Theorem 3.8. *Suppose that the conditions* (3.7)*,* (3.10) *and* (3.11)*, and Assumptions 1, 2, 3 and 4 are satisfied. Then Algorithm 3.3 terminates with an* $(\varepsilon, \delta_1 + \delta_2)$*-optimal solution of* (3.3) *or "infeasible" after a finite number of nodes.*

Proof. First of all note that under the assumptions of Theorem 3.8 the assumptions of Lemma 3.7 are satisfied, i.e., Algorithm 3.2 terminates finitely. Suppose that Algorithm 3.3 does not terminate. Then it produces at least one infinite nested sequence of nodes $\mathcal{F}_k = X_k \times Y_k^0 \times Y_k^S \times Z_k$, a sequence of solutions $(\tilde{\alpha}^k, \tilde{x}^k, \tilde{y}^{0,k}, \tilde{y}^{S,k}, \tilde{z}^k)$ of (3.5) over $\mathcal{F}_k$ found during the last iterations of Algorithm 3.2, and a sequence of parameters N^k. Note that the relaxation has to be feasible for every node, otherwise the node would be pruned and the sequence $\mathcal{F}_k$ ends finitely.

We show that there exists an iteration $K \in \mathbb{N}$ such that $(\tilde{\alpha}^K, \tilde{x}^K, \tilde{y}^{0,K}, \tilde{y}^{S,K}, \tilde{z}^K)$ is a $(\delta_1+\delta_2)$-feasible solution of (3.3). By the conditions (3.7) and (3.10) there exists an iteration $k_0 \in \mathbb{N}$ such that for all iterations $k \geq k_0$ the strict inequalities

$$|C(x, y^0, y^S, z) - \check{C}_k(x, y^0, y^S, z)| < \varepsilon$$

and

$$\|G(x, y^0, y^S, z) - \check{G}_k(x, y^0, y^S, z)\|_\infty < \delta_1$$

hold for all $(x, y^0, y^S, z) \in \mathcal{F}_k$. Then after k_0 nodes the only constraint which can be violated is

$$F^\ell\big(x, y^0, N\big) - \delta_1 \leq y^S \leq F^u\big(x, y^0, N\big) + \delta_1.$$

However, by condition (3.11), there is an iteration $k_1 \in \mathbb{N}$ such that this condition holds for all solutions $(\tilde{\alpha}^k, \tilde{x}^k, \tilde{y}^{0,k}, \tilde{y}^{S,k}, \tilde{z}^k)$ of (3.5) with $k \geq k_1$ produced by Algorithm 3.2. Moreover, Algorithm 3.2 increases N until the condition (3.6) holds with N^k at the solution. Hence, solution $(\tilde{x}^K, \tilde{y}^{0,K}, \tilde{y}^{S,K}, \tilde{z}^K)$ with $K = \max\{k_0, k_1\}$ is a $(\delta_1 + \delta_2)$-feasible solution of (3.3). Thus, the upper bound $\mathcal{U}$ will be updated if $C(\tilde{x}^K, \tilde{y}^{0,K}, \tilde{y}^{S,K}, \tilde{z}^K) < \mathcal{U}$ holds and the node $\mathcal{F}_K$ will be fathomed, because

$$\tilde{\alpha}^K \geq C\big(\tilde{x}^K, \tilde{y}^{0,K}, \tilde{y}^{S,K}, \tilde{z}^K\big) - \varepsilon \geq \mathcal{U} - \varepsilon$$

is satisfied for $K \geq k_0$. That is, no further branching occurs and the algorithm does not produce an infinite sequence of nodes and therefore terminates finitely.

It remains to show that the output of the algorithm is correct. Suppose Algorithm 3.3 terminates with upper bound $\mathcal{U} = \infty$. This only happens if every leaf of the branch-and-bound tree was fathomed, because the relaxations are infeasible. Since the leaf nodes define a partition of the feasible set and the relaxations are infeasible, the original problem has to be as well.

Suppose the algorithm terminates with an optimal solution $(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z})$. By construction of the algorithm and Lemma 3.4, it is clear that the solution is $(\delta_1 + \delta_2)$-feasible for (3.3). We distinguish two cases:

1. The feasible set of (3.3) is empty, i.e., $C^* = \infty$.
2. There is an optimal solution of (3.3) with optimal value $C^* < \infty$.

In the first case, clearly $C(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z}) - \varepsilon \leq C^*$ holds and $(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z})$ is ε-optimal. In the second case, let $\mathcal{F}_k = X_k \times Y_k^0 \times Y_k^S \times Z_k$ denote all nodes of the branch-and-bound tree which are fathomed due to $\alpha^k \geq \mathcal{U}^k - \varepsilon$ with optimal solution value α^k of the relaxation and current upper bound $\mathcal{U}^k$. Then $\bigcup_k \mathcal{F}_k$ defines a partition of the feasible set and $\min_k \alpha^k$ is a lower bound for C^*. With $C(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z}) \leq \mathcal{U}^k$ we can derive

$$C(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z}) - \varepsilon \leq \mathcal{U}^k - \varepsilon \leq \alpha^k$$

and therefore the inequality

$$C(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z}) - \varepsilon \leq \min_k \ \alpha^k \leq C^*,$$

is true, i.e., $(\bar{x}, \bar{y}^0, \bar{y}^S, \bar{z})$ is ε-optimal. □

Theorem 3.8 shows that Algorithm 3.3 works. On this basis, we apply Algorithms 3.2 and 3.3 to the example of stationary gas transport, which will yield Algorithms 4.1 and 4.2, in the next chapter. Moreover, we present first numerical results on a small network at the end of the next chapter. Finally, in Chapter 6 we present our implementation and more numerical results on a larger network.

Adaptivity and Feasibility

Algorithm 3.3 has the desired advantages over the two approaches discussed in Section 3.3. That is, it incorporates choosing N only as big as necessary in a single branch-and-bound tree. However, to be really able to call it adaptive in the choice of N the algorithm should also feature the possibility to decrease N. Especially if the functions F^ℓ and F^u are based on one-step methods as discussed in Chapter 2, we can reduce the computational time through using larger step sizes if possible. Therefore, we discuss how we can include decreasing N in Algorithms 3.2 and 3.3 in this section. Moreover, as mentioned before, changing N directly influences the feasibility notion in our algorithm. Thus, we address this topic here too.

First of all note that it is not just a mere theoretical observation that the (δ-)feasibility of solutions depends on N and thus at which point in time the solution is found. Our implementation for the example of stationary gas transport is based on Lemma 2.23 and currently uses (for technical reasons) a global discretization for each pipeline. That is, once the discretization is refined this finer discretization is used in every node of the branch-and-bound tree which is processed afterwards. There we can actually observe solutions which are not δ_1-feasible when feasibility is checked with the final discretization, although they were once declared δ_1-feasible (with respect to a coarser discretization) and stored as best solution; see Line 11 of Algorithm 3.3. For the test set of 4227 instances and our default parameters as will be described in Section 6.3 this happened 25 times.

The logical consequence is to use N only locally and also allow to decrease N in Algorithm 3.2. That is, to choose the least N necessary such that the bound condition (3.6) holds; see Lines 6 to 8. However, this entails several consequences. First and foremost, the proof of Lemma 3.7 and thus also the proof of Theorem 3.8 is based on the property that N only has to be increased finitely often until condition (3.6) is satisfied for all $(x, y^0) \in \tilde{X}_k \times \tilde{Y}_k^0$ on all nodes $\mathcal{F}_k$. Moreover, decreasing N can lead to an inconsistent feasibility notion, too, that is, previously δ_1-infeasible solutions can be δ_1-feasible after decreasing N.

Furthermore, we remark that we have to keep cuts which were added prior to decreasing N even if they cut off solutions which would be δ-feasible with respect to the decreased parameter N, otherwise this might lead to cycling of Algorithm 3.2. Suppose that we decrease N^1 to N^2 and would remove cuts which we previously added with respect to N^1. Then it might happen that we find a solution again which was previously cut off and requires to increase N^2 back to N^1. Removing cuts again, could then lead to cycling of Algorithm 3.2. We can obviously avoid this problem by not removing any cut which is not redundant. However, note that we actually could remove previous under- and overestimators $\check{F}^\ell$ and F^u after changing N if we only increase N.

Next, we consider the example of stationary gas transport again. Recall that Lemma 2.23 provides lower bounds on the step sizes such that the explicit midpoint method and the trapezoidal rule provide convex lower and upper bounds $P^\ell(p_{out}, q, N)$ and $P^u(p_{out}, q, N)$. Thus, we have to start the spatial branch-and-bound algorithm with a discretization which satisfies the bounds on the step sizes. Since these bounds depend on the maximal ratio $\nu_c = \frac{c\,q}{A\,p}$ and thus the lower bound of p_{out}, we can use a coarser discretization if the lower bound of p_{out} has been increased, e.g., via branching. Though then a feasible solution found after decreasing N, might have been infeasible with respect to the initial discretization. Therefore, to define a consistent δ_1-feasibility check in Algorithm 3.2 the least $N \geq N^0$ such that

$$\left\|F^u(\tilde{x}, \tilde{y}^0, N) - F^\ell(\tilde{x}, \tilde{y}^0, N)\right\|_\infty \leq \delta_2$$

holds should be used for the feasibility check, but with $N \geq N^0$ where N^0 is the initial parameter N^0 the algorithm was started with. However, note that we can use smaller values of N for separating infeasible solutions than for testing feasibility. If we combine increasing N with the feasibility check in Algorithm 3.2 (Lines 6 and following), then we need not increase N until (3.6) is satisfied, if we can already detect infeasibility of the solution before.

As mentioned above, if we allow to decrease N in Algorithm 3.2 then the proof of Lemma 3.7 does not hold any more. Thus, we have to assume that Algorithm 3.2 terminates finitely to show that Algorithm 3.3 terminates finitely. That is, if we replace Assumption 4 with the assumption that Algorithm 3.2 terminates finitely, then Theorem 3.8 still holds.

To summarize the discussion in this section: We can extend our approach to choose N adaptively by allowing N to be decreased in Algorithm 3.2. But to define a consistent feasibility check, we cannot use N smaller than the initial parameter N^0

when testing feasibility. However, for the separation of infeasible solutions we do not have to use the same parameter N as for testing feasibility. That is, we can also use smaller values of N for separation than N^0. Moreover, the construction of F^ℓ and F^u has to be designed in a way such that Algorithm 3.2 always terminates finitely not only under the assumption that N is kept fixed.

CHAPTER 4

Stationary Gas Transport

The aim of this chapter is to apply the generic spatial branch-and-bound algorithm for solving mixed-integer ODE constrained optimization problems, which we developed in the previous chapter, to our recurring example – stationary gas transport. The particular differential equation we consider is given by the stationary isothermal Euler equation (1.8), which was introduced in Section 1.1.

We will start this chapter with a short literature review and then introduce the components of gas networks and the models we use for them in Section 4.2. Afterwards in Section 4.3, we study the relaxation for the differential equations based on Lemma 2.23 and adjust Algorithm 3.2 to the setting of this chapter. In Section 4.4, we finally apply our spatial branch-and-bound Algorithm 3.3 and show that the assumptions of Theorem 3.8 are satisfied, which proves that the algorithm can be applied to the example of stationary gas transport and terminates after a finite number of iterations. After discussing some possible changes to model and how to adapt the spatial branch-and-bound algorithm to them in Section 4.5, we conclude this chapter with some preliminary numerical results in Section 4.6.

Parts of this chapter have been published in the article [56] which is joint work with Marc E. Pfetsch and Stefan Ulbrich. In particular, Algorithm 4.1 and the Propositions 4.5 and 4.6 already appeared in similar form there.

4.1 Literature Review

Gas transport in general is an example for a class of flow problems in networks for which many open questions and mathematical challenges exist; see Hante et al. [58]. Thus, gas transport, stationary as well as instationary, is a very active field of research and one can easily fill surveys about gas transport longer than this chapter, see Ríos-Mercado and Borraz-Sánchez [114], and also books about stationary gas transport, see Koch et al. [82]. Hence, we mainly refer to these publications and the references therein.

One commonly used idea for the optimization of gas transport problems involving discrete variables is to rely on state-of-the-art solvers for mixed integer linear problems (MIP). Therefore, nonlinear models respectively functions are often transformed into MIPs by either piecewise linearization or piecewise linear relaxation. This idea is applied to the stationary case, for example, by Martin et al. [96], and Geißler and coworkers [44, 45]. Articles that use this idea for the instationary case are, for example, Mahlke et al. [94], Domschke et al. [29], and Gugat et al. [52]. Thereby, the MIPs are usually constructed with a priori error tolerances or refined a posteriori, which leads to *multi-tree* approaches. Moreover, solving the MIPs is often part of some higher level algorithm, e.g., in [29] MIPs and nonlinear problems are solved alternatingly, in [44] MIPs are solved repeatedly in an alternating direction method, and in [52] a MIP is solved for each time step of an instantaneous control approach.

This idea is also applied in two further articles by Gugat et al. [53] and Schmidt et al. [128]. In both articles, global decomposition approaches for mixed-integer nonlinear problems are described. In the master problems of the decomposition methods a MIP relaxation of the original MINLP is solved and the subproblem computes a feasible solution of the nonlinear equality constraints which is close to the optimal MIP solution. The solutions of the subproblems are then used to iteratively refine the MIP relaxations. In the first article relaxations are constructed for functions $f\colon \mathbb{R} \to \mathbb{R}$ which are strictly monotonic, strictly concave or convex, and have a bounded first derivative. In the second article equality constraints with univariate Lipschitz continuous functions are considered and the relaxations are constructed using either known or approximated Lipschitz constants. Moreover, the function evaluations in the subproblems may be only approximate.

These two articles are related to our approach as follows. The properties which are used to construct the relaxations of the functions f in the first article, are similar to the properties which we exploit to construct relaxations of the gas flow in Section 4.3. Furthermore, both articles apply their method to a stationary ODE constrained gas transport problem very similar to ours. Since in every iteration of the decomposition methods a MIP is solved to global optimality, these are *multi-tree* methods, while our spatial branch-and-bound Algorithm 3.3 works as a *single-tree* method. However, both approaches are more limited than ours in the sense that they can only be applied to gas networks which are trees, while our method can be applied to general gas networks involving cycles. This is because their relaxations only work for one dimensional functions. To the best of our knowledge there are only some ideas using triangulations in the PhD thesis of Mathias Sirvent [142] and very preliminary results in Schmidt et al. [127], which do not yet overcome this problem successfully.

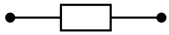
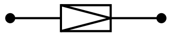
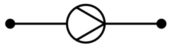
The existence of solutions of the stationary Euler equations for an ideal gas, i.e., compressibility factor $z \equiv 1$, on networks has been studied by Gugat et al. [51]. Schmidt et al. [129] present solutions for specific cases on single pipelines. Moreover, Gugat et al. [54] show the existence of stationary states on networks for real gases. Additionally, they provide a comparison of the analytical solution with the Weymouth equation (1.9).

4.2 Modeling Stationary Gas Networks

Gas networks mainly consist of pipelines for the transport of gas, but also include a variety of additional components. Most important are compressor stations to compensate for the pressure loss induced by the transport, valves to route the gas through the network, and control valves to reduce the pressure for the transition from a high-pressure transport network to a low-pressure distribution network. For example, the German high-pressure gas transport network comprises almost 40 000 kilometers of pipelines and a compressor station every 100 to 200 kilometers; see the web page gas network operators [150].

As it is usually done in the literature, we model a gas transport network as a directed graph $\mathcal{D} = (\mathcal{V}, \mathcal{A})$. The nodes $\mathcal{V}$ are *entries*, *exits* and *junctions* of the network. The entries and exits, hereinafter also called sources and sinks, are points where gas can be fed into or withdrawn from the network. In practice, entries and exits can be producers and customers of gas as well as gas storages, interconnections with other networks or connections across borders. Note that over time the distinction between entries and exits is not fixed, e.g., a storage has to be filled before one can withdraw gas from the storage. Nevertheless, for the stationary setting, which we consider here, this cannot occur. Thus, the distinction between entries, exits and junctions yields a partition of the nodes into the entries $\mathcal{V}_+$, exits $\mathcal{V}_-$ and the remaining nodes $\mathcal{V}_0$. The arcs $\mathcal{A}$ represent the different network components. Apart from the previously mentioned *pipelines*, *compressor stations* and *(control) valves* our model includes *short cuts* and *resistors*. Short cuts are essentially very short pipelines where the pressure drop is assumed to be negligible. Short cuts are sometimes part compressor stations, where they are used to connect different components. Resistors form a surrogate model for pressure loss which one does not want to or cannot model in detail. An example for the former are gas preheaters, which sometimes appear in a compressor station, because their behavior is highly nonlinear. Examples for the latter are dirt in the pipelines, strong curvature of pipelines, or filtering devices.

Table 4.1. Overview on arc types, their respective subsets of the arcs $\mathcal{A}$ and their symbol in figures.

Arc type	Set	Symbol in figures
pipeline	$\mathcal{A}^{pi}$	
short cut	$\mathcal{A}^{sc}$	
valve	$\mathcal{A}^{va}$	
resistor	$\mathcal{A}^{re}$	
control valve	$\mathcal{A}^{cv}$	
compressor station	$\mathcal{A}^{cs}$	

We will now introduce the variables and constraints of the model. We start with those which are common for all arc types and then the specific constraints for different arc types. Thereby, we use indices to denote which variables are associated with nodes or arcs as well as the dependency of parameters.

Note. The formulas presented in the following use the *International System of Units (SI)*. Our implementation does not use these units for all variables, but using them renders the formulas more readable.

The main variables in our model are pressure variables p_v measured in Pa for all nodes $v \in \mathcal{V}$ and mass flow variables q_a in kg/s for all arcs $a \in \mathcal{A}$. These variables have lower and upper bounds $\underline{p}_v$, $\overline{p}_v$ and $\underline{q}_a$, $\overline{q}_a$, respectively. Note that $\underline{q}_a < 0$ is possible and denotes flow in the opposite direction of the arc a. Furthermore, we consider a vector $q^{\pm} \in \mathbb{R}^{\mathcal{V}}$ of inflows and outflows of gas into and out of the network. The in- and outflows have to be balanced and determine the partition into entries, exits, and junctions, i.e., $q^{\pm}$ has to satisfy $\sum_{v \in \mathcal{V}} q_v^{\pm} = 0$, inflow is given by $q_v^{\pm} > 0$ at the entries $v \in \mathcal{V}_+$, outflow is $q_v^{\pm} < 0$ at the exits $v \in \mathcal{V}_-$, and $q_v^{\pm} = 0$ for the remaining nodes $v \in \mathcal{V}_0$. Then the basic constraints of our model are given by the flow conservation constraints on the network and the variable bounds

$$
\begin{aligned}
&\sum_{a \in \delta^+(v)} q_a - \sum_{a \in \delta^-(v)} q_a = q_v^{\pm} && \forall v \in \mathcal{V}, \\
&\underline{p}_v \le p_v \le \overline{p}_v && \forall v \in \mathcal{V}, \\
&\underline{q}_a \le q_a \le \overline{q}_a && \forall a \in \mathcal{A}.
\end{aligned} \tag{4.1}
$$

Thereby, $\delta^+(v)$ denotes the outgoing arcs of node v and $\delta^-(v)$ denotes the incoming arcs of v, i.e.,

$$\delta^+(v) := \{a \in \mathcal{A} : a = (v,u) \text{ for some } u \in \mathcal{V}\},$$
$$\delta^-(v) := \{a \in \mathcal{A} : a = (u,v) \text{ for some } u \in \mathcal{V}\}.$$

We will now go through the arc types in the order of Table 4.1 and introduce the corresponding models. The model for pipelines is given by the differential equation (1.8), which we derived in Section 1.1. For compressor stations we are using an idealized model in pressure and mass flow variables derived by Hiller and Walther [69]. The remaining models are taken from Chapter 6 of the book by Koch et al. [82].

4.2.1 Pipelines

Consider a *pipeline* $a = (u,v) \in \mathcal{A}^{pi}$, in the following also called *pipe* for short. Recall from Section 1.1 the stationary isothermal Euler equation

$$\partial_x p_a(x) \left(1 - \frac{c^2 q_a^2}{A_a^2 p_a(x)^2}\right) = -\frac{\lambda_a c^2}{2 D_a A_a^2} \frac{q_a |q_a|}{p_a(x)} - \frac{g}{c^2} \sigma_a \, p_a(x), \quad x \in [0, L_a], \tag{1.8}$$

which describes the pressure of gas flowing through a pipeline, and the short notation

$$\partial_x p_a(x) = \varphi_{\sigma_a,a}\big(p_a(x), q_a\big), \quad x \in [0, L_a]. \tag{1.11}$$

As in Section 2.3 we mainly consider pipelines without height differences, i.e., $\sigma_a = 0$, and require that

$$\frac{c|q_a|}{A_a p_a(x)} \leq \nu_c \tag{4.2}$$

holds with $\nu_c \in (0,1)$, such that $\partial_p \varphi$ is bounded. In Section 4.5 we discuss how to adapt our algorithmic approach to cope with height differences. Due to the assumption $\sigma_a = 0$, we omit the index and use $\varphi_a = \varphi_{0,a}$.

We couple the differential equation with the pressure variables at the nodes by

$$p_u = p_a(0), \quad p_v = p_a(L_a)$$

and enforce condition (4.2) via the linear inequalities

$$0 \leq \nu_c A_a \, p_u + c \, q_a, \quad 0 \leq \nu_c A_a \, p_v - c \, q_a. \tag{4.3}$$

Table 4.2. Variables and parameters of a pipeline $a = (u, v) \in \mathcal{A}^{pi}$.

Symbol	Description	Unit
p_u, p_v	pressure variables at the nodes u and v	Pa
q_a	flow variable for pipeline a	$\mathrm{kg\,s^{-1}}$
$p_a(x)$	pressure at state x; solves ODE (1.8)	Pa
A_a	cross-sectional area of the pipeline	$\mathrm{m^2}$
D_a	diameter of the pipeline	m
λ_a	friction coefficient of the pipeline	1
L_a	length of the pipeline	m
σ_a	slope of the pipeline	1

Note that these two inequalities are sufficient to represent condition (4.2), since the pressure is decreasing in the direction of the flow. That is, we only have to require that the condition is satisfied at the end of the pipeline where the flow leaves the pipeline; see also Corollary 2.17.

Table 4.2 shows the variables and parameters associated with a pipeline. The complete set of constraints for each pipeline $a = (u, v) \in \mathcal{A}^{pi}$ is

$$\begin{aligned}
&\partial_x p_a(x) = \varphi_a\big(p_a(x), q_a\big), \quad x \in [0, L_a],\\
&p_u = p_a(0),\ p_v = p_a(L_a),\\
&0 \le \nu_c A_a\, p_u + c\, q_a,\\
&0 \le \nu_c A_a\, p_v - c\, q_a.
\end{aligned} \tag{4.4}$$

4.2.2 Short Cuts

Short cuts $a = (u, v) \in \mathcal{A}^{sc}$ have no other variables or parameters than the pressure variables p_u, p_v associated with the incident nodes and the mass flow q_a. The mass flow is only constrained by its lower and upper bounds and the pressure variables have to be equal

$$p_u = p_v. \tag{4.5}$$

4.2.3 Valves

As mentioned before, *valves* are used for routing the gas flow through the network and through compressor stations. They are also used to detach parts of the network, e.g., for maintenance. A valve can be open or closed. If it is closed, there can be no gas flow and the pressures variables at both ends of the valve are decoupled. If

Table 4.3. Variables of a valve $a = (u, v) \in \mathcal{A}^{va}$.

Symbol	Description	Unit
p_u, p_v	pressure variables at the nodes u and v	Pa
q_a	flow variable for valve a	$\mathrm{kg\,s^{-1}}$
z_a	binary variable; represents state of the valve	1

the valve is open, gas can flow in both directions within the given flow bounds and the pressure loss is negligible, i.e., it acts as a short cut and the pressure variables at both ends have to be equal.

For a valve $a = (u, v) \in \mathcal{A}^{va}$ we introduce a binary variable $z_a \in \{0, 1\}$, where $z_a = 1$ represents an open valve and $z_a = 0$ represents a closed valve. With this binary variable we can model the flow constraint and enforce the equality $p_u = p_v$ for an open valve by the following inequalities

$$\begin{aligned} &\underline{q}_a\, z_a \leq q_a \leq \overline{q}_a\, z_a, \\ &(\underline{p}_u - \overline{p}_v)(1 - z_a) \leq p_u - p_v, \\ &p_u - p_v \leq (\overline{p}_u - \underline{p}_v)(1 - z_a). \end{aligned} \tag{4.6}$$

4.2.4 Resistors

We distinguish two types of *resistors*: Linear resistors, where the pressure loss only depends on the flow direction, and nonlinear resistors, where the pressure loss also depends on the amount of flow.

For a linear resistor $a = (u, v)$ the pressure loss is given by the discontinuous function

$$p_u - p_v = \begin{cases} \xi_a & \text{if } q_a > 0, \\ 0 & \text{if } q_a = 0, \\ -\xi_a & \text{if } q_a < 0, \end{cases} \tag{4.7}$$

with a fixed pressure drop $\xi_a > 0$. We approximate this discontinuous function by the piecewise linear function

$$p_u - p_v = \begin{cases} \xi_a & \text{if } q_a > q_\varepsilon, \\ \xi_a \frac{q_a}{q_\varepsilon} & \text{if } -q_\varepsilon \leq q_a \leq q_\varepsilon, \\ -\xi_a & \text{if } q_a < -q_\varepsilon, \end{cases} \tag{4.8}$$

where we use the constant $q_\varepsilon = \frac{1}{3600}\,\mathrm{m^3\,s^{-1}}\rho_0$ as suggested by Geißler et al. [45]. Thereby ρ_0 is the density of the gas under normal conditions. To model this function

Table 4.4. Variables and parameters of a linear resistor $a = (u, v) \in \mathcal{A}^{re}$.

Symbol	Description	Unit
p_u, p_v	pressure variables at the nodes u and v	Pa
q_a	flow variable for resistor a	$\mathrm{kg\,s^{-1}}$
$z_a^{\varepsilon-}$	binary indicator variable for $q_a \leq -q_\varepsilon$	1
$z_a^{\varepsilon+}$	binary indicator variable for $q_a \geq q_\varepsilon$	1
q_a^0	additional variable used for representation of (4.8)	1
ξ_a	fixed pressure loss	Pa
q_ε	constant flow threshold; $q_\varepsilon = \frac{1}{3600}\,\mathrm{m^3\,s^{-1}}\rho_0$	$\mathrm{kg\,s^{-1}}$

with linear constraints we introduce three additional variables; two binary variables $z_a^{\varepsilon-}$, $z_a^{\varepsilon+} \in \{0, 1\}$ and a continuous variable $q_a^0 \in [-1, 1]$; see also Table 4.4. We couple these variables with the mass flow by the inequalities

$$\begin{aligned} &z_a^{\varepsilon-} + z_a^{\varepsilon+} + q_a^0 \leq 1, \\ &z_a^{\varepsilon-} + z_a^{\varepsilon+} - q_a^0 \leq 1, \\ &\underline{q}_a z_a^{\varepsilon-} + q_\varepsilon z_a^{\varepsilon+} + q_\varepsilon q_a^0 - q_a \leq 0, \\ &q_\varepsilon z_a^{\varepsilon-} - \overline{q}_a z_a^{\varepsilon+} - q_\varepsilon q_a^0 + q_a \leq 0. \end{aligned} \tag{4.9}$$

That is, $z_a^{\varepsilon-}$ and $z_a^{\varepsilon+}$ are indicator variables whether the flow is less than $-q_\varepsilon$ or greater than q_ε, respectively. If $z_a^{\varepsilon-} = 1$, we obtain $z_a^{\varepsilon+} = 0$, $q_a^0 = 0$ and $\underline{q}_a \leq q_a \leq -q_\varepsilon$. Analogously, if $z_a^{\varepsilon+} = 1$, then $z_a^{\varepsilon-} = 0$, $q_a^0 = 0$ and $q_\varepsilon \leq q_a \leq \overline{q}_a$ holds true. Otherwise, if $z_a^{\varepsilon-} = z_a^{\varepsilon+} = 0$, the equality $q_\varepsilon q_a^0 = q_a$ holds. Then (4.8) can be expressed by the equality

$$p_u - p_v = -\xi_a z_a^{\varepsilon-} + \xi_a z_a^{\varepsilon+} + \xi_a q_a^0. \tag{4.10}$$

Next, we present the model for nonlinear resistors $a = (u, v) \in \mathcal{A}^{re}$. A nonlinear resistor is uniquely determined by its diameter D_a and the *drag factor* ζ_a. Unlike before, the pressure decrease not only depends on the direction of the flow, but also

Table 4.5. Variables and parameters of a nonlinear resistor $a = (u, v) \in \mathcal{A}^{re}$.

Symbol	Description	Unit
p_u, p_v	pressure variables at the nodes u and v	Pa
q_a	flow variable for resistor a	$\mathrm{kg\,s^{-1}}$
ζ_a	drag factor of the resistors	1
D_a	diameter of the resistors	m

on the amount of the flow. The change in pressure is described by

$$p_u - p_v = \begin{cases} \beta_a q_a^2\, p_u^{-1} & \text{if } q_a > 0, \\ 0 & \text{if } q_a = 0, \\ -\beta_a q_a^2\, p_v^{-1} & \text{if } q_a < 0, \end{cases} \tag{4.11}$$

where the coefficient β_a is given by

$$\beta_a = \frac{8}{\pi^2 D_a^4}\, \zeta_a\, R_s\, T_m\, z_m.$$

Here, we use again the specific gas constant R_s, the constant mean gas temperature T_m, the formula (1.6) for the mean pressure p_m, and the formula (1.4) to compute the mean compressibility factor $z_m = z(p_m, T_m)$.

Finally, the dependency of the pressure difference with the flow direction can be expressed by the nonlinear equation

$$p_u^2 - p_v^2 + |p_u - p_v|(p_u - p_v) = 2\beta_a |q_a| q_a. \tag{4.12}$$

To see this observe that (4.11) implies $\operatorname{sgn}(p_u - p_v) = \operatorname{sgn}(q_a)$. Consider the case $p_v = p_u$ first. This obviously implies $2\beta_a |q_a| q_a = 0$, i.e., $q_a = 0$. Otherwise, if inequality $p_u > p_v$ holds, then by (4.12) we get

$$2\beta_a |q_a| q_a = p_u^2 - p_v^2 + |p_u - p_v|(p_u - p_v) = 2p_u(p_u - p_v) > 0.$$

That is, we have $q_a > 0$ and the first case of (4.11) holds. Analogously we can see that the third case of (4.11) is satisfied if $p_u < p_v$.

4.2.5 Control Valves

A *control valve* $cv = (u, v) \in \mathcal{A}^{cv}$ is an unidirectional network element used to reduce the pressure from node u to v. A control valve can be *active* or *closed*. If it is closed, then the control valve acts like a normal valve, i.e., there is no gas flow and the pressure variables are decoupled. If the control valve is active, then only nonnegative flow is possible and the pressure is reduced from node u to v; thereby the possible pressure reduction is bounded by

$$0 \le \underline{\Delta}_{cv} \le p_u - p_v \le \overline{\Delta}_{cv}.$$

Furthermore, if the control valve is active, the inlet and outlet pressures have to satisfy technical limits $\underline{p}_{cv} \le p_u$ and $p_v \le \overline{p}_{cv}$.

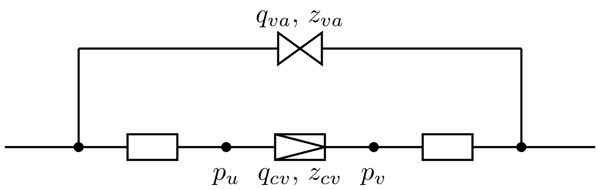


Figure 4.1. Schematic plot of a control valve station with a bypass and two resistors.

Analogously to valves, we model the state of the control valve by introducing a binary variable $z_{cv} \in \{0, 1\}$ with $z_{cv} = 1$ representing the active state and $z_{cv} = 0$ representing the closed state. With this variable, the complete model for a control valve is

$$
\begin{aligned}
&0 \leq q_{cv} \leq \overline{q}_{cv}\, z_{cv},\\
&(\underline{p}_u - \overline{p}_v)(1 - z_{cv}) + \underline{\Delta}_{cv}\, z_{cv} \leq p_u - p_v,\\
&p_u - p_v \leq (\overline{p}_u - \underline{p}_v)(1 - z_{cv}) + \overline{\Delta}_{cv}\, z_{cv},\\
&\underline{p}_{cv}\, z_{cv} \leq p_u,\\
&p_v \leq \overline{p}_{cv}\, z_{cv} + \overline{p}_v(1 - z_{cv}),
\end{aligned}
\tag{4.13}
$$

and a summary of variables and parameters of a control valve can be seen in Table 4.6. Note that the last two constraints only have to be added to the model if the technical limits $\underline{p}_{cv}$, $\overline{p}_{cv}$ are tighter than the corresponding variable bounds at nodes u and v.

Typically control valves are combined with other elements as follows; see Figure 4.1 for a schematic plot of a so-called *control valve station*. Due to the *Joule-Thomson effect* the gas temperature decreases, when the gas expands while passing through a control valve. To avoid big changes in temperature and even damage of the valve, gas heaters are installed before the actual control valve. Moreover, they are sometimes combined with measurement devices which can cause additional pressure loss. If necessary, these effects are modeled by resistors before and after the control valve. Furthermore, since control valves have a fixed working direction, there is often an additional *bypass valve* to make flow in the reverse direction possible. A bypass valve $va \in \mathcal{A}^{va}$ is just a usual valve with the additional constraint that the bypass cannot be open if the control valve is active

$$z_{cv} + z_{va} \leq 1. \tag{4.14}$$

Table 4.6. Variables and parameters of a control valve $cv = (u, v) \in \mathcal{A}^{cv}$.

Symbol	Description	Unit
p_u, p_v	pressure variables at the nodes u and v	Pa
q_{cv}	flow variable for control valve cv	$\mathrm{kg\,s^{-1}}$
z_{cv}	binary variable; represents state of the control valve	1
$\underline{\Delta}_{cv}$, $\overline{\Delta}_{cv}$	minimal/maximal pressure decrease in active state	Pa
$\underline{p}_{cv}$, $\overline{p}_{cv}$	technical pressure limits for the active state	Pa

4.2.6 Compressor Stations

Compressor stations are the most complicated network elements. They are used to increase the pressure level, such that the gas can be transported through the network. They usually comprise several compressor machines, drives, filtering devices, and gas coolers. These elements are often combined by complex variable routing of the gas flow such that compressor machines can be used in so-called *configurations*. For example, if there are two compressor machines, they can (sometimes) be used in *parallel* or in *serial*. Moreover, also only one compressor machine or none might be used.

Detailed physical models of compressor stations and machines feature several nonlinear equations; e.g., see Schmidt et al. [129] or Chapter 2 in [82]. In particular, the operating ranges of turbo compressors are usually described by *characteristic diagrams* in terms of adiabatic head and volumetric flow. Thereby, the adiabatic head depends nonlinearly on the ratio of input and output pressure. Furthermore, detailed models also have to take changes in temperature due to compression and cooling into account. Since the focus of this thesis is not on detailed compressor models, we use a simplified model computed by Hiller and coworkers [69, 154].

The model bounds the mass flow, input and output pressure by a polyhedron and is available for compressor machines as well as configurations. The polyhedron combines lower and upper bounds on the variables, minimal and maximal pressure increase, and minimal and maximal relative pressure increase. These bounds only have to hold if the compressor machine is running and are derived in the following way. After sampling the characteristic diagram of a compressor machine, the samples are transformed into a cloud of points in the (q, p^{in}, p^{out})-space (by sampling p^{in}). Afterwards, points which violate technical limits, e.g., if the drive cannot deliver the required power, are discarded. Then the bounds on the variables and (relative) pressure increase are defined by the remaining points. The polyhedra for compressor machines can then be combined into polyhedra for configurations. The constraints

Table 4.7. Variables and parameters of a compressor station $cs = (u, v) \in \mathcal{A}^{cs}$.

Symbol	Description	Unit
p_u, p_v	pressure variables at the nodes u and v	Pa
q_{cs}	flow variable for compressor station cs	$\mathrm{kg\,s^{-1}}$
z_{cs}	binary variable; represents state of the compressor station	1
p_c^{in}, p_c^{out}	pressure variables for configuration $c \in \mathcal{C}_{cs}^{cf}$	Pa
q_c	flow variable for configuration $c \in \mathcal{C}_{cs}^{cf}$	$\mathrm{kg\,s^{-1}}$
z_c	binary variable; represents if configuration $c \in \mathcal{C}_{cs}^{cf}$ is active	1
$\underline{\Delta}_c$, $\overline{\Delta}_c$	minimal/maximal pressure increase of $c \in \mathcal{C}_{cs}^{cf}$	Pa
$\underline{\varepsilon}_c$, $\overline{\varepsilon}_c$	minimal/maximal relative pressure increase of $c \in \mathcal{C}_{cs}^{cf}$	1

for this model (for both compressor machines and configurations) are given by

$$\begin{aligned}
&\underline{q} \le q \le \overline{q},\\
&\underline{p}^{in} \le p^{in} \le \overline{p}^{in},\\
&\overline{p}^{in} \le p^{out} \le \overline{p}^{out},\\
&\underline{\Delta} \le p^{out} - p^{in} \le \overline{\Delta},\\
&\underline{\varepsilon}\, p^{in} \le p^{out} \le \overline{\varepsilon}\, p^{in},
\end{aligned} \tag{4.15}$$

with $\underline{\Delta} \ge 0$, $\underline{\varepsilon} \ge 1$ and $\underline{q} > 0$.

A more detailed model would be given by the convex hull of the points in the (q, p^{in}, p^{out})-space. However, in general the convex hull has many facets. Hence, Walther et al. [154] propose an algorithm to choose a small number of the facets such that they define a polyhedron which is close to the convex hull w.r.t. the volumes of the polyhedra. Relying on the precomputed inequalities by Walther et al., we have implemented both versions of this model, i.e., with and without the additional facets, and will investigate their influence on computational results in Section 6.3.3.

Consider a compressor station $cs = (u, v) \in \mathcal{A}^{cs}$ and let $\mathcal{C}_{cs}^{cf}$ be the set of its configurations. Besides the usual variables p_u, p_v and q_{cs}, we use a binary variable z_{cs} to represent if the compressor station is active or inactive. Furthermore, we introduce additional variables p_c^{in}, p_c^{out}, q_c and $z_c \in \{0, 1\}$ for all configurations $c \in \mathcal{C}_{cs}^{cf}$, because we want to avoid turning the constraints (4.15) and, in particular, the additional facets on and off by using Big Ms; see Table 4.7 for an overview on variables and parameters of a compressor station. With the binary variables z_c we represent which configuration is used, if the compressor station is active. If $z_c = 1$, we couple the pressure and mass flow variables, that is, $p_c^{in} = p_u$, $p_c^{out} = p_v$ and $q_c = q_{cs}$. Otherwise, if $z_c = 0$, the variables are decoupled. If the station is inactive, i.e., $z_{cs} = 0$, we set $q_{cs} = 0$ and $z_c = 0$ for all configurations $c \in \mathcal{C}_{cs}^{cf}$. Then the set of constraints for the compressor station is

$$
\begin{aligned}
& 0 \leq q_{cs} \leq \overline{q}_{cs}\, z_{cs}, && \\
& \textstyle\sum_{c \in \mathcal{C}_{cs}^{cf}} z_c = z_{cs}, && \\
& -\overline{q}_c\,(1 - z_c) \leq q_{cs} - q_c \leq (\overline{q}_{cs} - \underline{q}_c)\,(1 - z_c) && \forall c \in \mathcal{C}_{cs}^{cf}, \\
& (\underline{p}_u - \overline{p}_c^{in})\,(1 - z_c) \leq p_u - p_c^{in} \leq (\overline{p}_u - \underline{p}_c^{in})\,(1 - z_c) && \forall c \in \mathcal{C}_{cs}^{cf}, \\
& (\underline{p}_v - \overline{p}_c^{out})\,(1 - z_c) \leq p_v - p_c^{out} \leq (\overline{p}_v - \underline{p}_c^{out})\,(1 - z_c) && \forall c \in \mathcal{C}_{cs}^{cf}, \\
& (q_c, p_c^{in}, p_c^{out}) \text{ satisfy (4.15) and additional facets} && \forall c \in \mathcal{C}_{cs}^{cf}.
\end{aligned}
\tag{4.16}
$$

Note that the mass flow variables for the configurations are not directly part of the flow conservation in (4.1), but indirectly through the coupling of q_{cs} and q_c if the corresponding configuration is active.

Similar to control valves the induced pressure decrease due to further elements in the compressor station is represented by resistors. Furthermore, also compressor stations can often be bypassed, which is again modeled by an additional valve $va \in \mathcal{A}^{va}$ with

$$z_{cs} + z_{va} \leq 1. \tag{4.17}$$

Figure 4.2 shows a schematic of a compressor station model with two configurations c_1 and c_2, and an inlet and outlet resistor.

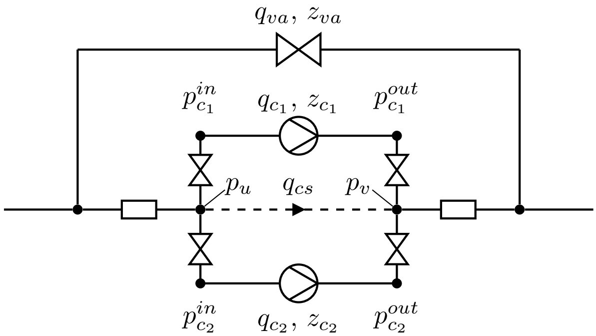


Figure 4.2. Schematic plot of a compressor station with configurations c_1 and c_2, a bypass, and two resistors.

4.2.7 Optimization Model for Stationary Gas Transport

Now that we have introduced the models of the network elements, we can summarize the model in the following optimization problem:

$$\begin{aligned} \min\ & C(p,q,z) \\ \text{s.t.}\ & G(p,q,z) \le 0, \\ & \partial_x p_a(x) = \varphi_a\big(p_a(x), q_a\big) && x \in [0, L_a],\ \forall\, a \in \mathcal{A}^{pi} \subseteq \mathcal{A}, \\ & p_u = p_a(0),\ p_v = p_a(L_a) && \forall\, a = (u,v) \in \mathcal{A}^{pi}, \\ & p \in P,\ q \in Q,\ z \in Z. \end{aligned} \tag{4.18}$$

The variables $p \in P$ comprise the pressure variables p_v for all nodes $v \in \mathcal{V}$ and the pressure variables p^{in}, p^{out} for the compressor station configurations. The flow variables $q \in Q$ include the mass flow variables q_a for all arcs $a \in \mathcal{A}$, the auxiliary variables q_a^0 for linear resistors, and the additional mass flow variables for the compressor station configurations. The variables $z \in Z \subseteq \{0,1\}^m$ are the binary variables for valves, linear resistors, control valves, and compressor stations. Finally, we have the function variables $p_a(x)$ for each pipe $a \in \mathcal{A}^{pi} \subseteq \mathcal{A}$ which have to satisfy the stationary isothermal Euler equation. The sets P and Q are given by the variable bounds in (4.1) and the bounds of the auxiliary variables as discussed where they were introduced. Except for the differential equations the constraint $G(p,q,z) \le 0$ represents the models for the different network elements, which we introduced above.

Note that (4.18) is a special case of the abstract problem (3.1). Here, the flow variables q correspond to the variables x in the abstract setting. The pressure variables at the end of pipelines correspond to y^0 and y^S, and the binary variables z, here, are integer or binary variables in (3.1). We have constraints $G \le 0$ in both problems and, moreover, the differential equations in (4.18) only define a coupling between the pressure and flow variables for a single pipe.

There are plenty options for the objective function $C(p,q,z)$. One possibility is $C \equiv 0$ which turns problem (4.18) into a feasibility problem to identify whether there is a feasible solution at all. Another possible objective is to find an energy efficient solution. Since we cannot compute the power consumption of the compressor stations with the model we use, we can instead minimize the number of running compressors. A third possible objective is to minimize the power lost due to transportation; therefore, consider a certain amount of gas which is transported through a pipeline $a = (u,v)$. In a stationary isothermal process the energy required to transport the gas from u to v is given by $p_u V_u - p_v V_v$, where V_u and V_v are the volume of the gas at u and v. Thus, the power required to transport the mass flow q through the network is given by taking the difference of the pressure times volumetric flow at the sources and sinks. Since the mass flow is proportional to the volumetric flow, we can minimize the pressure times the mass flow at the sources and sinks as a proxy for the power loss, i.e., we minimize the linear function

$$C(p,q,z) = \sum_{a=(u,v)\in\mathcal{A}} (p_u - p_v)\, q_a = \sum_{v\in\mathcal{V}} \Bigg(\sum_{a=(v,w)\in\mathcal{A}} q_a - \sum_{a=(u,v)\in\mathcal{A}} q_a \Bigg) p_v = \sum_{v\in\mathcal{V}} q_v^{\pm}\, p_v.$$

Remark 4.1. Except for the differential equations, the models presented in this section can directly be treated by state-of-the-art MINLP solvers like ANTIGONE [102], BARON [123], COUENNE [11], or SCIP [40]. These solvers are based on spatial branch-and-bound to ensure global optimality and rely on well-known techniques, such as McCormick inequalities [99], outer-approximation by Duran and Grossmann [30], the reformulation-linearization technique by Sherali and coworkers [137, 138, 139], or the αBB method by Adjiman and coworkers [3, 4] to derive convex or even linear relaxations of the original problem. Moreover, often polyhedral convex envelopes for edge-convex or edge-concave functions are constructed, see for example Meyer and Floudas [101] or Tardella [147], and underestimators for specific terms are used, e.g., see Liberti and Pantelides [89] for convex envelopes of monomials of odd degrees, or Tawarmalani et al. [149].

Remark 4.2. Note that Gugat et al. [54] showed the existence of stationary states in passive, connected networks, i.e., connected networks of pipelines only. They prove that in the absence of variable bounds there is a unique solution if the pressure at one source is fixed and the inflows are sufficiently small. Furthermore, they provide analytical solutions for the stationary isothermal Euler equations (1.8) with slope $\sigma = 0$. Thus, we could replace the ODE constraints (4.4) by using the analytical solution $P(p(0), q) = p(L)$. Note that this step is analogous to the reformulation of problem (3.1) to problem (3.3). However, evaluating this analytical solution requires numerically solving the inverse of some function, e.g., by Newton's method. For example, in the case of an ideal gas one has to evaluate the Lambert-W function; see Gugat et al. [51]. Hence, the analytical solution is not suited for standard MINLP techniques.

4.3 LP-Relaxation for Gas Flow on Pipelines

In order to apply the adaptive spatial branch-and-bound framework from Chapter 3 to problem (4.18), we have to define relaxations of (4.18) equivalently to the relaxations (3.4) and (3.5) of problem (3.1). Since state-of-the-art MINLP solvers can treat all models presented in the previous section except for the differential equations by well-known techniques, we only discuss how to compute valid linear relaxations of (4.4) for a linear programming (LP) based branch-and-bound method in this section. Furthermore, we derive a problem specific version of Algorithm 3.2 for constructing the relaxations of the gas flow.

Recall from Chapter 2 the definition of $P^\ell(p,q,N) = p^\ell_N$ and $P^u(p,q,N) = p^u_N$ through the evaluation of the explicit midpoint

$$p^\ell_0 = p, \qquad p^\ell_i = p^\ell_{i-1} - h\,\varphi\big(p^\ell_{i-1} - \tfrac{h}{2}\varphi(p^\ell_{i-1},q),q\big) \qquad \forall i \in [N], \tag{2.18}$$

and the implicit trapezoidal rule

$$p^u_0 = p, \qquad p^u_i = p^u_{i-1} - \tfrac{h}{2}\left[\varphi(p^u_{i-1},q) + \varphi(p^u_i,q)\right] \qquad \forall i \in [N], \tag{2.19}$$

with initial values $p^\ell_0 = p^u_0 = p$, mass flow $q \geq 0$, and N discretization steps. Consider a pipeline $a = (u,v) \in \mathcal{A}^{pi}$. Due to Lemma 2.23 we can utilize the functions P^ℓ and P^u to define a relaxation of

$$\begin{aligned} &\partial_x p_a(x) = \varphi_a\big(p_a(x),q_a\big), \quad x \in [0,L_a],\\ &p_u = p_a(0),\ p_v = p_a(L_a) \end{aligned}$$

as a special case of (3.4) as follows. If the direction of the flow q_a is fixed, that is, variable q_a is either nonnegative or nonpositive, and N sufficiently big, then

$$P^\ell_a(p_{out},|q_a|,N_a) \leq p_{in} \leq P^u_a(p_{out},|q_a|,N_a) \tag{4.19}$$

defines a relaxation for a single pipeline, where p_{in} and p_{out} are chosen such that they coincide with the nodes where the gas enters and leaves the pipeline. That is, if $q_a \geq 0$, we have $p_{in} = p_u$ and $p_{out} = p_v$. Note, that we use P^ℓ_a and P^u_a with index a, since the right-hand sides φ_a of the ODEs can differ in length, diameter, and so forth. Otherwise, if the flow direction is not fixed, we ignore the differential equations. Once all flow directions are fixed, for example, through branching on the flows w.r.t. $q_a = 0$, the relaxation defined this way, satisfies Assumption 2, in particular, the functions P^ℓ_a and P^u_a converge to the exact solution of the differential equation for $N_a \to \infty$; see Lemma 2.23.

We point out that P^ℓ and P^u are given by an iterative scheme instead of a single explicit formula. Hence, standard techniques to derive convex relaxations are not directly applicable. Instead we designed an adaptive approach analogously to Algorithm 3.2, which incorporates branching on the flow directions and has the advantage that the evaluation of P^ℓ and P^u is only needed "on demand". Furthermore, the number of discretization steps need not satisfy condition (3.6) up front for the whole domain, but the condition is enforced in the course of the branch-and-bound process for specific points. To construct linear under- and overestimators of P^ℓ and P^u, we use step sizes satisfying the requirement of Lemma 2.23, such that P^ℓ and P^u are convex. Thus, we can underestimate P^ℓ by outer-approximation and P^u admits a vertex polyhedral concave envelope.

4.3.1 Linear Underestimators

In Section 2.3, we have seen that $P^\ell \colon U \times \mathbb{N} \to \mathbb{R}$ is convex and continuously differentiable on the domain

$$U = \{(p,q) \in \mathbb{R}^2 \,:\, \underline{p} \le p \le \overline{p},\ 0 \le \underline{q} \le q \le \overline{q},\ 0 \le \nu_c A\, p - c\, q\}$$

if N is sufficiently big. Thus, for $(\tilde{p}, \tilde{q}) \in U$ the inequality

$$P^\ell(\tilde{p}, \tilde{q}, N) + \nabla P^\ell(\tilde{p}, \tilde{q}, N)^\top \begin{pmatrix} p - \tilde{p} \\ q - \tilde{q} \end{pmatrix} \le P^\ell(p, q, N)$$

holds for all $(p,q) \in U$. Hence, once the flow direction of a pipe is fixed, we can approximate the lower bound $P_a^\ell(p_{out}, q_a, N_a) \le p_{in}$ arbitrarily well by iteratively adding gradient cuts

$$P_a^\ell(\tilde{p}_{out}, \tilde{q}_a, N_a) + \nabla P_a^\ell(\tilde{p}_{out}, \tilde{q}_a, N_a)^\top \begin{pmatrix} p_{out} - \tilde{p}_{out} \\ q_a - \tilde{q}_a \end{pmatrix} \le p_{in} \tag{4.20}$$

for different pairs $(\tilde{p}_{out}, \tilde{q}_a)$; see Duran and Grossmann [30].

Note that although gradient cuts w.r.t. N_a discretization steps, might not be a valid underestimator for $P_a^\ell(p, q, N_a')$ with another number of discretization steps N_a', it still is a valid underestimator for the exact ODE solution. This implies that we can keep gradient cuts added to the relaxation, when changing the discretization.

Since P^ℓ is given by (2.18), we cannot directly compute its partial derivatives $\partial_p P^\ell$ and $\partial_q P^\ell$. Instead consider the function $p^{em}(p, q, h)$, which performs one step of (2.18), as defined in the proof of Lemma 2.23. Then with the approximations p_i^ℓ interpreted as functions of (p,q) for $i \in [N]$ and $p_0^\ell(p,q) = p$, we get

$$p_i^\ell(p,q) = p^{em}\big(p_{i-1}^\ell(p,q), q, h\big) = p_{i-1}^\ell(p,q) - h\, \varphi\big(p_{i-1}^\ell(p,q) - \tfrac{h}{2}\varphi(p_{i-1}^\ell(p,q), q), q\big)$$

for $i \in [N]$. Again, differentiating $p_i^\ell(p,q)$ yields $\partial_p\, p_0^\ell(p,q) = 1$, $\partial_q\, p_0^\ell(p,q) = 0$ and

$$\begin{aligned}
&\partial_p\, p_i^\ell(p,q) = \partial_p p^{em}\big(p_{i-1}^\ell(p,q), q, h\big)\ \partial_p\, p_{i-1}^\ell(p,q), \\
&\partial_q\, p_i^\ell(p,q) = \partial_p p^{em}\big(p_{i-1}^\ell(p,q), q, h\big)\ \partial_q\, p_{i-1}^\ell(p,q) + \partial_q p^{em}\big(p_{i-1}^\ell(p,q), q, h\big)
\end{aligned}$$

for $i \in [N]$. That is, we can iteratively evaluate the derivatives of P^ℓ and thus construct linear underestimators.

4.3.2 Linear Overestimators

While in general the so-called *concave envelope*, i.e., the smallest concave overestimator, of a function is neither linear nor easy to construct, in our case P^u admits a *vertex polyhedral concave envelope* on U, i.e., the concave envelope of P^u is affine linear and determined by P^u evaluated at the vertices of U. In the following, we will first collect some known results from the literature and then apply these to P^u to show how we can construct its concave envelope.

It seems that the notion of convex/concave envelopes was first introduced by Kleibohm [81], who defined a "konvexe Unterfunktion" as the greatest convex function, which is smaller than the function at hand. There are several well-known equivalent definitions by Falk [33], Falk and Soland [35], and in particular the following definition due to Rockafellar [116].

Definition 4.3. *Let $X \subseteq \mathbb{R}^d$ and consider a function $f\colon X \to \mathbb{R}$. The* convex envelope *of f is the function $\operatorname{vex}_X[f]\colon \operatorname{conv}(X) \to \mathbb{R}$ defined by*

$$\operatorname{vex}_X[f](x) := \inf\Big\{\sum_{i=1}^{d+1} \lambda_i f(x_i) \;:\sum_{i=1}^{d+1} \lambda_i x_i = x,\ \mathbb{1}^\top \lambda = 1,\ \lambda \geq 0,\ x_1, \ldots, x_{d+1} \in X\Big\},$$

where $\mathbb{1}$ is the vector of ones with appropriate dimension. Analogously, the concave envelope $\operatorname{cave}_X[f]\colon \operatorname{conv}(X) \to \mathbb{R}$ *of f is given by*

$$\operatorname{cave}_X[f](x) := \sup\Big\{\sum_{i=1}^{d+1} \lambda_i f(x_i) \;:\sum_{i=1}^{d+1} \lambda_i x_i = x,\ \mathbb{1}^\top \lambda = 1,\ \lambda \geq 0,\ x_1, \ldots, x_{d+1} \in X\Big\}.$$

Furthermore, a subset $G \subseteq X$ is called generating set of the convex or concave envelope of f if $\operatorname{vex}_G[f] = \operatorname{vex}_X[f]$ *or* $\operatorname{cave}_G[f] = \operatorname{cave}_X[f]$ *holds, respectively.*

Note that since we want to apply the following results to construct the concave envelope of P^u, we restrict the presentation to concave envelopes. However, because $\operatorname{cave}_X[f] = -\operatorname{vex}_X[-f]$ holds, the results also apply to the convex envelope.

For general functions it is a NP-hard problem to construct the concave envelope; for example, see the complexity results by Crama [27] or by Kalantari and Rosen [77]. In fact, it is even NP-hard to evaluate the concave envelope at a single point: Tardella [147] showed that evaluating the concave envelope of a function $f\colon \{0,1\}^d \to \mathbb{R}$ with the property $f(x) = f(\mathbb{1} - x)$ for all x at the point $\frac{1}{2}\mathbb{1}$ is as hard as maximizing f over $\{0,1\}^d$. Moreover, for every function $g\colon \{0,1\}^{d-1} \to \mathbb{R}$ we can define a function $f\colon \{0,1\}^d \to \mathbb{R}$ satisfying condition $f(x) = f(\mathbb{1} - x)$ through

$$f(y,1) = f(\mathbb{1} - y, 0) = g(y)$$

for all $y \in \{0,1\}^{d-1}$. Since maximizing g over $\{0,1\}^{d-1}$ is NP-hard, evaluating concave envelopes is NP-hard in general.

However, in particular cases it is both easy to construct and evaluate the concave envelope. The concave envelope of $f\colon X \to \mathbb{R}$ has a finite generating set G, if and only if it is *polyhedral* (e.g., see Rockafellar [116]), i.e., there exists a finite set of indices I and pairs $(\alpha_i, \beta_i) \in \mathbb{R}^d \times \mathbb{R}$ for $i \in I$, such that the concave envelope of f is

$$\operatorname{cave}_X[f](x) = \min_{i \in I}\ \alpha_i^\top x + \beta_i.$$

Rikun [113] showed that for the special case of continuously differentiable functions on a polytope X, the concave envelope is polyhedral if and only if it is *vertex polyhedral*, i.e., the vertices $\operatorname{vert}(X)$ are a generating set. Note that there are different terms for this property, e.g., Tawarmalani et al. [149] call a function *concave-extendable* from the vertices if the vertices are a generating set. Several conditions for this property are known, e.g., Rikun [113] showed that the vertices of a box are a generating set for multilinear functions, whereas Falk and Hoffman [34] showed that the vertices of a polytope are a generating set for the concave envelope if f is convex. More general, Tardella [147] showed that $\operatorname{vert}(X)$ is a generating set if f is *edge-convex*, i.e., f is convex on all line segments which are parallel to some edge of X. Note that if X is a box, f is edge-convex, if it is convex w.r.t. every single variable.

Let $X \subset \mathbb{R}^d$ be a full-dimensional polytope and consider a function $f\colon X \to \mathbb{R}$ with the generating set $\operatorname{vert}(X)$. Let $\Delta \subseteq X$ be a *d-simplex* with $\operatorname{vert}(\Delta) \subseteq \operatorname{vert}(X)$, i.e., the set Δ is the convex hull of $d+1$ affine independent vertices of X. We denote the affine linear function defined through interpolating f at the vertices of Δ with $L_{\Delta,f}\colon \mathbb{R}^d \to \mathbb{R}$, that is, $L_{\Delta,f}(\bar{x}) = f(\bar{x})$ for all $\bar{x} \in \operatorname{vert}(\Delta)$. The following result shows how to determine whether $L_{\Delta,f}$ coincides with $\operatorname{cave}_X[f]$ on Δ. Note that similar results can be found in Tardella [147], Meyer and Floudas [101], and Tawarmalani et al. [149].

Proposition 4.4. *Let $X \subset \mathbb{R}^d$ be a full-dimensional polytope and let* $\operatorname{vert}(X)$ *be a generating set for the concave envelope of $f\colon X \to \mathbb{R}$. Consider a d-simplex $\Delta \subseteq X$ with* $\operatorname{vert}(\Delta) \subseteq \operatorname{vert}(X)$*. Then $L_{\Delta,f}\colon \mathbb{R}^d \to \mathbb{R}$ defines a facet of the concave envelope of f over X, i.e., $L_{\Delta,f}(x) = \operatorname{cave}_X[f](x)$ for all $x \in \Delta$, if and only if the inequality $f(x_i) \le L_{\Delta,f}(x_i)$ is satisfied for all vertices $x_i \in \operatorname{vert}(X) = \{x_1, \ldots, x_k\}$.*

Proof. By assumption $\operatorname{vert}(X)$ is a generating set of the concave envelope, that is, by definition we have

$$\operatorname{cave}_X[f](x) = \max\left\{\sum_{i=1}^{k} \lambda_i f(x_i) \,:\, \sum_{i=1}^{k} \lambda_i x_i = x,\ \mathbb{1}^\top \lambda = 1,\ \lambda_i \geq 0\right\}.$$

The dual problem of this is

$$\begin{aligned} \min\ & x^\top \alpha + \beta \\ \text{s.t.}\ & f(x_i) \leq x_i^\top \alpha + \beta, \quad \forall i \in [k] \\ & \alpha \in \mathbb{R}^d,\ \beta \in \mathbb{R}. \end{aligned}$$

The dual problem shows that all affine linear functions which define a facet of the concave envelope have to majorize f at the vertices.

For the reverse direction, assume that $L_{\Delta,f}$ satisfies $f(x_i) \leq L_{\Delta,f}(x_i)$ for all vertices $x_i \in \operatorname{vert}(X)$. Then obviously $\operatorname{cave}_X[f] \leq L_{\Delta,f}$ holds on X. Suppose that $L_{\Delta,f}$ does not define a facet of the concave envelope. Hence, there exists $x \in \Delta$ with $\operatorname{cave}_X[f](x) < L_{\Delta,f}(x)$. But, since $f(x_i) = \operatorname{cave}_X[f](x_i) = L_{\Delta,f}(x_i)$ for all vertices $x_i \in \operatorname{vert}(\Delta)$, this contradicts $\operatorname{cave}_X[f]$ being concave. Thus $L_{\Delta,f}$ defines a facet of the concave envelope. □

Moreover, Tawarmalani et al. [149] derived a complete characterization for vertex polyhedral concave envelopes through *triangulations*. A triangulation T of a full-dimensional polytope X is a set of d-simplices such that the vertices of the simplices are vertices of X, the intersection of two simplices is either empty or a face of both simplices, the intersection of two simplices has no interior point, and the union of the simplices is the polytope X. Again, consider a function $f\colon X \to \mathbb{R}$. Tawarmalani et al. have proven that there exists a triangulation T of a full-dimensional polytope X such that

$$\operatorname{cave}_X[f](x) = \min_{\Delta \in T}\ L_{\Delta,f}(x)$$

if and only if $\operatorname{vert}(X)$ is a generating set of the concave envelope.

Finally, we apply these results to construct the concave envelope of P^u on U. Suppose that $U \subset \mathbb{R}^2$ is not empty or a singleton. Then the intersection of the half space defined by $0 \leq \nu_c A\, p - c\, q$ with the box $[\underline{p}, \overline{p}] \times [\underline{q}, \overline{q}]$ given by the variable bounds has five possible shapes, which are depicted in Figure 4.3. Using a sufficiently large number of discretization steps N, such that Lemma 2.23 holds, P^u is convex on U. Hence, $\operatorname{vert}(U)$ is a generating set of the concave envelope and there exists a

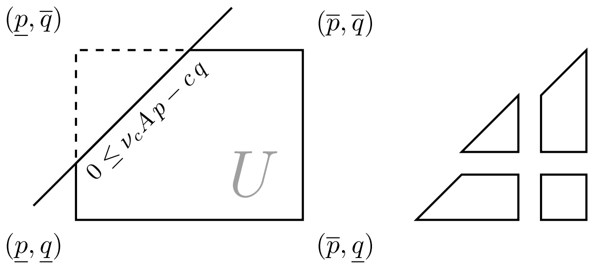


Figure 4.3. If the domain U of P^ℓ and P^u is not empty or a singleton, it has five possible shapes depicted in this figure.

triangulation T of U such that

$$\mathrm{cave}_U[P^u](p,q) = \min_{\Delta \in T} \; L_{\Delta,P^u}(p,q) \tag{4.21}$$

holds. Since U has 3, 4, or 5 vertices, the triangulation consists of 1 to 3 simplices and we can represent the concave envelope by at most three inequalities of the type

$$p_{in} \le L_{\Delta,P^u}(p_{out}, q)$$

with $\Delta \in T$. For example, consider $\Delta = \{(\underline{p}, \underline{q}),\, (\overline{p}, \underline{q}),\, (\overline{p}, \overline{q})\}$. In this case we have

$$\begin{aligned} L_{\Delta,P^u}(p,q) = P^u(\overline{p}, \underline{q}, N) &+ \left[P^u(\underline{p}, \underline{q}, N) - P^u(\overline{p}, \underline{q}, N)\right] \frac{p - \overline{p}}{\underline{p} - \overline{p}} \\ &+ \left[P^u(\overline{p}, \overline{q}, N) - P^u(\overline{p}, \underline{q}, N)\right] \frac{q - \underline{q}}{\overline{q} - \underline{q}}. \end{aligned}$$

4.3.3 Relaxation Algorithm

In this section, we adjust Algorithm 3.2 to the adaptive construction of the relaxation for P^ℓ and P^u. Therefore, in Algorithm 4.1 we proceed similar to Algorithm 3.2.

We initialize the branch-and-bound algorithm with the minimal number of discretization steps N_a, such that the conditions of Lemma 2.23 are satisfied for all pipes $a \in \mathcal{A}^{pi}$. For instance, with $\nu_c = 0.4$ we use N_a discretization steps such that inequality $\frac{L_a}{N_a} \le 4.925 \frac{D_a}{\lambda_a}$ holds. Then when called in a node of the branch-and-bound tree, we choose an LP-relaxation of the ODE constraints. In the root node, we ignore the ODE constraints and use the variable bounds as initial relaxation. In every other node, we use the relaxation of the parent node. Next, we solve the convex or LP-relaxation equivalent to problem (3.5). Recall, in the relaxation we replace the constraints $G(p,q,z) \le 0$ by $\check{G}(p,q,z) \le 0$, and if C is nonconvex we add an auxiliary variable α, the constraint $\check{C}(p, q, z) \leq \alpha$ and minimize α. Therefore, we assume that the convex underestimators $\check{G}$ and $\check{C}$ of G and C (if the objective function is nonconvex) are given. If the relaxation is infeasible, then the node of the branch-and-bound tree can be cut off. Otherwise, if the relaxation is feasible, then we get a solution $(\tilde{p}_u, \tilde{p}_v, \tilde{q}_a)$ for each pipe $a \in \mathcal{A}^{pi}$. For the ease of presentation, let these solutions be given as triples $(\tilde{p}_{in}, \tilde{p}_{out}, \tilde{q}_a)$, where $\tilde{p}_{in}$ is the pressure at the node where gas flows into the pipe, $\tilde{p}_{out}$ accordingly the other pressure value, and $\tilde{q}_a \geq 0$.

As the next step, we compute $P_a^\ell(\tilde{p}_{out}, \tilde{q}_a, N_a)$ and $P_a^u(\tilde{p}_{out}, \tilde{q}_a, N_a)$ for all $a \in \mathcal{A}^{pi}$. If the differences do not satisfy (3.6), we increase the number of discretization steps N_a, and recompute P_a^ℓ and P_a^u until they do; see Lines 7 and 8. Afterwards, we check whether the inequality

$$P_a^\ell(\tilde{p}_{out}, \tilde{q}_a, N_a) \leq \tilde{p}_{in} \leq P_a^u(\tilde{p}_{out}, \tilde{q}_a, N_a) \tag{4.22}$$

is satisfied with feasibility tolerance δ_1 for all pipes. If all triples are δ_1-feasible, Algorithm 4.1 returns the current solution to the branch-and-bound process. In the case that at least one triple is not feasible, we pick the pipe a with the largest deviation of $\tilde{p}_{in}$ from the closest bound $P_a^\ell(\tilde{p}_{out}, \tilde{q}_a, N_a)$, or $P_a^u(\tilde{p}_{out}, \tilde{q}_a, N_a)$; see Line 11. If the flow direction of this pipe is not fixed, we return q_a as branching candidate w.r.t. $q_a = 0$ to the branch-and-bound process; see Line 13. Note that this step is particular to our context here, since the construction of under- and overestimators for P^ℓ and P^u, which was described above, requires a fixed flow direction. Hence, this step is an addition to Algorithm 3.2. Otherwise, if the flow direction is already fixed, we try to cut off the solution as discussed above. To this end, we distinguish three different cases of infeasibility, as shown in Figure 4.4.

Let $(\tilde{p}_{in}, \tilde{p}_{out}, \tilde{q}_a)$ be a δ_1-infeasible solution of (4.22). In the first case, $\tilde{p}_{in}$ is larger than the concave envelope of P_a^u over the domain $\tilde{U}_a$ given by the variable bounds in the current node of the branch-and-bound tree, and the constraint $0 \leq \nu_c A\, p_{out} - c\, q$; see Line 15. In this case, we can choose if we either add all of the linear inequalities (4.21), which define the concave envelope, or choose at least one of them, which cuts off the current solution; see Section 4.3.2. In the second case, we have

$$P_a^u(\tilde{p}_{out}, \tilde{q}_a, N_a) \leq \tilde{p}_{in} \leq \operatorname{cave}_{\tilde{U}_a}[P_a^u](\tilde{p}_{out}, \tilde{q}_a, N_a),$$

i.e., we cannot cut off the current solution with the concave envelope; see Line 17. Instead, we have to resolve the infeasibility by branching w.r.t. to either p_{in}, p_{out} or q_a. In the last case (Line 20), when $\tilde{p}_{in}$ is less than $P_a^\ell(\tilde{p}_{out}, \tilde{q}_a, N_a)$, we can use

Algorithm 4.1 Adaptive convex relaxation of gas flow

Input: Node of branch-and-bound tree $\tilde{P} \times \tilde{Q} \times \tilde{Z}$, δ_1, $\delta_2 > 0$ and $N \in \mathbb{N}^{\mathcal{A}^{pi}}$.

Output: δ_1-feasible solution of (4.19), “infeasible” or “branch”.

Choose LP-relaxation of ODE constraints:
In the root node take the box $P \times Q$, else use the relaxation of the parent node.

For $k = 1, 2, \ldots$ **do**
  Solve the convex relaxation of (4.18) on node $\tilde{P} \times \tilde{Q} \times \tilde{Z}$.
  If the relaxation is feasible **then**
    let $\left(\tilde{\alpha}^k, \tilde{p}^k, \tilde{q}^k, \tilde{z}^k\right)$ be the solution and for each pipe $a \in \mathcal{A}^{pi}$ denote the solution with $\left(\tilde{p}^k_{in}, \tilde{p}^k_{out}, \tilde{q}^k_a\right)_a$.
    For all $a \in \mathcal{A}^{pi}$ **do**
      While $\left|P^u_a\left(\tilde{p}^k_{out}, \tilde{q}^k_a, N_a\right) - P^\ell_a\left(\tilde{p}^k_{out}, \tilde{q}^k_a, N_a\right)\right| > \delta_2$ **do**
        increase N_a.
    If all $\left(\tilde{p}^k_{in}, \tilde{p}^k_{out}, \tilde{q}^k_a\right)_a$ are δ_1-feasible for (4.22) **then**
      return the solution $\left(\tilde{\alpha}^k, \tilde{p}^k, \tilde{q}^k, \tilde{z}^k\right)$.
    Choose “most violated” pipe $a \in \mathcal{A}^{pi}$, i.e.,
$$a \in \arg\max_{a \in \mathcal{A}^{pi}} \max\left\{\tilde{p}^k_{in} - P^u_a\left(\tilde{p}^k_{out}, \tilde{q}^k_a, N_a\right),\ P^\ell_a\left(\tilde{p}^k_{out}, \tilde{q}^k_a, N_a\right) - \tilde{p}^k_{in}\right\}.$$
    If $\underline{q}_a < 0 < \overline{q}_a$ **then**
      suggest branching w.r.t. $q_a = 0$ to fix orientation of flow on pipe a and
      return “branch”.
    If $\tilde{p}^k_{in} > \operatorname{cave}_{\tilde{U}_a}[P^u_a]\left(\tilde{p}^k_{out}, \tilde{q}^k_a, N_a\right)$ **then**
      add (at least one separating inequality of) the concave envelope of P^u_a to the relaxation,
    else if $\tilde{p}^k_{in} > P^u_a\left(\tilde{p}^k_{out}, \tilde{q}^k_a, N_a\right)$ **then**
      suggest branching w.r.t. to either p_{in}, p_{out}, or q_a and
      return “branch”,
    else if $\tilde{p}^k_{in} < P^\ell_a\left(\tilde{p}^k_{out}, \tilde{q}^k_a, N_a\right)$ **then**
      add a gradient cut of the form (4.20) to the relaxation,
  else
    return “infeasible”.

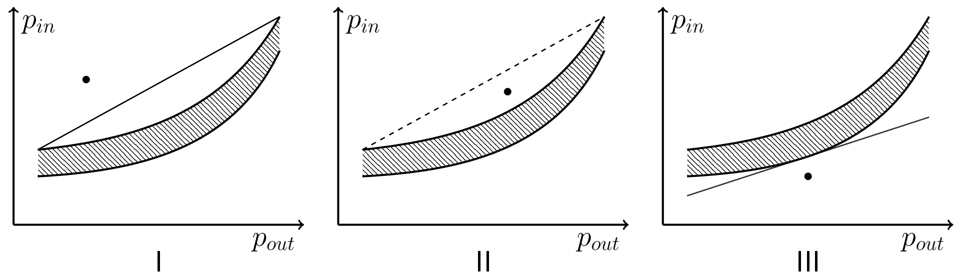


Figure 4.4. For a pair (p_{in}, p_{out}) and fixed mass flow rate q there are the three different cases of infeasibility of constraint (4.22). The feasible region is hatched and the three cases (from left to right) are as follows: p_{in} is greater than the concave envelope of P^u; p_{in} is greater than P^u, but cannot be cut off by the concave envelope; p_{in} is less than the lower bound and infeasibility can be resolved by adding a gradient cut.

the convexity of P_a^ℓ and cut off the solution with a gradient cut

$$P_a^\ell(\tilde{p}_{out}, \tilde{q}, N_a) + \nabla P_a^\ell(\tilde{p}_{out}, \tilde{q}_a, N_a)^\top \begin{pmatrix} p_{out} - \tilde{p}_{out} \\ q_a - \tilde{q}_a \end{pmatrix} \leq p_{in},$$

see Section 4.3.1.

In the first and last case, we then iterate and solve the relaxation again. In the second case, Algorithm 4.1 stops and instructs the branch-and-bound process to branch. After the convex relaxation algorithm terminates with a solution, we carry on like in the spatial branch-and-bound Algorithm 3.3.

In order to apply Theorem 3.8, which shows that our spatial branch-and-bound algorithm terminates finitely, we have to show that Algorithm 4.1 satisfies Assumption 4.

Proposition 4.5. *Algorithm 4.1 terminates after a finite number of iterations.*

Proof. We show that Assumption 4 holds for Algorithm 4.1, that is, that the algorithm terminates finitely if it keeps N fixed. If this assumption holds, then by Lemma 3.7 shows the statement. Thus, suppose that the vector of the numbers of discretization steps $N \in \mathbb{N}^{\mathcal{A}^{pi}}$ stays constant during the execution of Algorithm 4.1, i.e., the condition

$$\left| P_a^u(p_{out}^k, q_a^k, N_a) - P_a^\ell(p_{out}^k, q_a^k, N_a) \right| < \delta_2$$

is satisfied for all produced solutions $(p^k_{in}, p^k_{out}, q^k_a)_a$ for all pipes $a \in \mathcal{A}^{pi}$ and all iterations k. Moreover, note that it suffices to only consider a single pipe. One can straightforwardly extend this proof to an arbitrary number of pipes.

Suppose that the algorithm does not terminate, that is, it produces an infinite sequence of points which are feasible for the convex relaxation but not δ_1-feasible for (4.22). Since the algorithm does not terminate, the orientation of flow already has been fixed. Hence, we can distinguish between input and output pressure. Moreover, the second case of infeasibility can not occur since the algorithm would also terminate in this case; see Figure 4.4 and Line 17 of the algorithm.

Let $(p^k_{in}, p^k_{out}, q^k_a)_{k\in\mathbb{N}}$ denote the sequence of solutions produced by the algorithm. We divide the iterations into sets corresponding to the two possible cases of infeasibility. That is, we denote with $\mathcal{O} \subseteq \mathbb{N}$ the set of iterations with $p^k_{in} > \operatorname{cave}_U[P^u_a](p^k_{out}, q^k_a, N_a)$ and with $\mathcal{L} = \mathbb{N} \setminus \mathcal{O}$ the set of iterations with $p^k_{in} < P^\ell(p^k_{out}, q^k_a, N_a)$; see Lines 15 and 20. We will show that both sets have to be finite and therefore the algorithm terminates after a finite number of iterations.

We first consider the subsequence $\mathcal{O}$. Let $\tilde{U}_a$ denote the domain of P^u_a given by the variable bounds of p_{out} and q in the current node of the branch-and-bound tree, and the constraint $0 \le \nu_c A\, p_{out} - c\, q$. Obviously, $\tilde{U}_a$ still has one of the five shapes depicted in Figure 4.3. Hence, as discussed above in Section 4.3.2 the concave envelope of P^u_a on $\tilde{U}_a$ is given by at most three linear inequalities. Thus, even if we choose to add only one of the defining inequalities at each iteration $k \in \mathcal{O}$, it only takes up to three iterations to fully add the concave envelope to the relaxation. Furthermore, since neither branching occurs (and thus $\tilde{U}_a$ does not change) nor N_a is increased, the concave envelope of P^u_a does not change during the course of the algorithm. That is, $\mathcal{O}$ contains at most three iterations (in general, at most three times the number of pipes).

Next, we show that the sequence $\mathcal{L}$ is finite, too. In every iteration $k \in \mathcal{L}$ we add an inequality of the form (4.20) to the relaxation. Since P^ℓ_a is convex and continuously differentiable, it is Lipschitz continuous on the compact set $\tilde{U}_a$. Hence, there is a radius r such that the inequality

$$0 \le P^\ell_a\big(p_{out}, q_a, N_a\big) - P^\ell_a\big(p^k_{out}, q^k_a, N_a\big) - \nabla P^\ell_a\big(p^k_{out}, q^k_a, N_a\big)^\top \begin{pmatrix} p_{out} - p^k_{out} \\ q_a - q^k_a \end{pmatrix} < \delta_1$$

holds for all $k \in \mathcal{L}$ and for all points $(p_{out}, q_a) \in B_r(p^k_{out}, q^k_a) \cap \tilde{U}_a$, where $B_r(p^k_{out}, q^k_a)$ is the open ball around (p^k_{out}, q^k) with radius r. That is, any point (p_{in}, p_{out}, q_a) with $(p_{out}, q_a) \in B_r(p^k_{out}, q^k_a) \cap \tilde{U}_a$, which satisfies the gradient cut for iteration k, is a δ_1-feasible solution for the lower bound in (4.22).

Now, suppose that $\mathcal{L}$ is not finite. Then since $\tilde{U}_a$ is compact, there exists a subset $(k_i)_{i\in\mathbb{N}} \subset \mathcal{L}$ such that the sequence $(p^{k_i}_{out}, q^{k_i}_a)_{i\in\mathbb{N}}$ converges to $(p^*_{out}, q^*_a) \in \tilde{U}_a$.

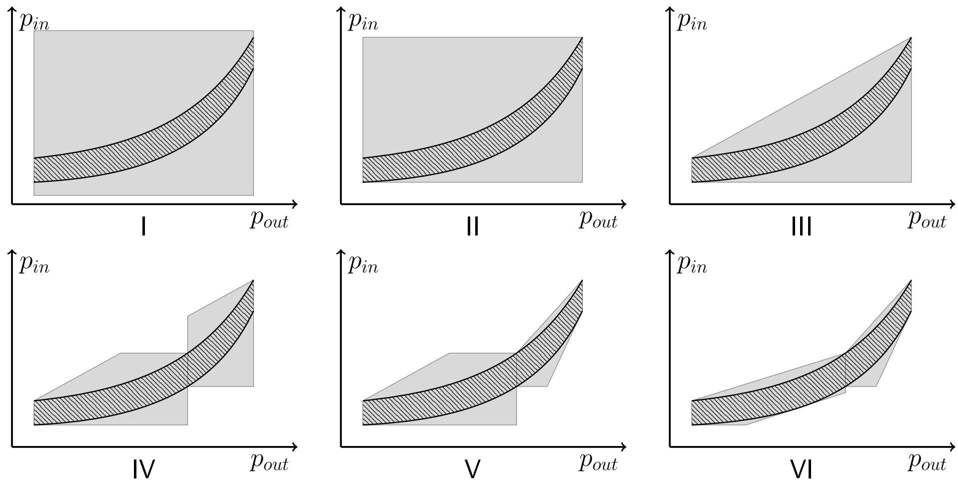


Figure 4.5. The figure shows an exemplary progression of the relaxation produced by repeatedly calling Algorithm 4.1. The feasible set defined by P^u and P^ℓ is hatched (with north west lines) and the relaxation is shaded gray. Thereby, the development of the relaxation is as follows: We start with the box defined by the variable bounds (I). Then after tightening the variable bounds (II), the concave overestimator is added (III). The bounds are tightened again after branching w.r.t. p_{out} (IV). Finally, the new overestimator and a gradient cut are added in the right node (V) and also in the left node (VI).

Let k_j be an element of the sequence, such that $(p^*_{out}, q^*_a) \in B_r(p^{k_j}_{out}, q^{k_j}_a)$ holds. Then, since the sequence converges to (p^*_{out}, q^*_a), there exists an index $n > j$ such that also $(p^{k_n}_{out}, q^{k_n}_a) \in B_r(p^{k_j}_{out}, q^{k_j}_a)$ holds. By the above argument this implies that $(p^{k_n}_{in}, p^{k_n}_{out}, q^{k_n}_a)$ is a δ_1-feasible solution of $P^\ell_a(p_{out}, q_a, N_a) \leq p_{in}$, i.e., the algorithm stops. Thus, $\mathcal{L}$ has to be finite. $\square$

4.4 Spatial Branch-and-Bound Algorithm for Stationary Gas Transport

We are now ready to apply the adaptive spatial branch-and-bound Algorithm 3.3 to problem (4.18) which yields Algorithm 4.2. We proceed analogously to the general framework. Given $\nu_c \in (0, 1)$, we initialize the algorithm with the minimal number of discretization steps $N_0 \in \mathbb{N}^{\mathcal{A}^{pi}}$ such that the conditions of Lemma 2.23 are met. The only differences are that we call Algorithm 4.1 instead of Algorithm 3.2 and the particular case of choosing the mass flow as branching variable in order to fix the flow direction; see Line 17. Note that we do not specify an order to choose the branching variables from the cases in Lines 14 to 18. In our implementation with the branch-and-bound framework SCIP [40, 132], we leave the choice of the branching variable to SCIP, because it includes various branching rules. However, note that SCIP itself branches on fractional binary oder integer variables first. For an overview and the importance of branching rules see, for example, Achterberg et al. [2] or Bonami et al. [16].

Theorem 3.8 shows that Algorithm 3.3 terminates after a finite number of iterations. To see that this also applies to Algorithm 4.2, we have to prove that condition (3.11) holds. That is, if the algorithm produces an infinite nested sequence of nodes and solutions $(\tilde{\alpha}^k, \tilde{p}^k, \tilde{q}^k, \tilde{x}^k)_{k\in\mathbb{N}}$ of the relaxation, then there has to exist an iteration $k_0 \in \mathbb{N}$ such that

$$\max\left\{p_{in}^k - P_a^u(p_{out}^k, q_a^k, N_a),\ P_a^\ell(p_{out}^k, q_a^k, N_a) - p_{in}^k\right\} \le \delta_1$$

holds for all pipes $a \in \mathcal{A}^{pi}$ and $k \ge k_0$. Figure 4.5 shows the exemplary development of the relaxation for a single pipe, which is produced by repeatedly calling Algorithm 4.1.

Proposition 4.6. *Let conditions* (3.7) *and* (3.10) *be true. Suppose that Algorithm 4.2 produces an infinite nested sequence of nodes. Then the solutions of the convex relaxation produced by Algorithm 4.1 satisfy condition* (3.11).

Proof. Again, it suffices to only consider a single pipe $a \in \mathcal{A}^{pi}$. Suppose that Algorithm 4.2 produces an infinite nested sequence of nodes. Since our first priority is to fix the direction of the flow, we can assume that q_a is restricted to nonnegative values. We denote the sequence of bounding boxes on pipe a with

$$\mathcal{F}_k = \left[\underline{p}_{in}^k, \overline{p}_{in}^k\right] \times \left[\underline{p}_{out}^k, \overline{p}_{out}^k\right] \times \left[\underline{q}_a^k, \overline{q}_a^k\right].$$

Further, let U_a^k denote the domain of P^u and P^ℓ in node k, and let $(p_{in}^k, p_{out}^k, q_a^k)$ be the last solution of the relaxation produced by Algorithm 4.1 for node k.

Note that the algorithm only returns $(p_{in}^k, p_{out}^k, q_a^k)$ with $P_a^\ell(p_{out}^k, q_a^k, N_a) - p_{in}^k > \delta_1$ if there is another pipe with δ_1-infeasible solution which cannot be separated by a cut. Otherwise, by construction of Algorithm 4.1, a gradient cut, which cuts off the current solution, would be added. Hence, it suffices to show that there exists $k_0 \in \mathbb{N}$ with $p_{in}^k - P_a^u(p_{out}^k, q_a^k, N_a) \le \delta_1$ for all $k \ge k_0$.

We will show that $k_0 \in \mathbb{N}$ exists such that the concave envelope of P_a^u cuts off all δ_1-infeasible points. Since (4.18) contains the constraint $0 \le \nu_c A\, p_{out} - c\, q_a$, we

Algorithm 4.2 Adaptive spatial branch-and-bound for stationary gas transport

Input: Problem (4.18), $\nu_c \in (0,1)$, $N = N_0 \in \mathbb{N}^{\mathcal{A}^{pi}}$, δ_1, $\delta_2 > 0$ and $\varepsilon > 0$.
Output: $(\varepsilon, \delta_1 + \delta_2)$-optimal solution (p^*, q^*, z^*) or "infeasible".

```
Upper bound U ← ∞
List of active nodes L ← {P × Q × Z}
While L ≠ ∅ do
    choose a node P̃ × Q̃ × Z̃ ∈ L and set L ← L \ {P̃ × Q̃ × Z̃}.
    Construct underestimators Č and Ǧ.
    Run Algorithm 4.1.
    If Algorithm 4.1 stops with a solution or "branch" then
        let (α̃, p̃, q̃, z̃) be the last solution found in Algorithm 4.1.
        If the solution is δ1-feasible for (4.22) then
            If (p̃, q̃, z̃) is δ1-feasible for G(p, q, z) ≤ 0 and C(p̃, q̃, z̃) < U holds then
                set U ← C(p̃, q̃, z̃) and (p*, q*, z*) ← (p̃, q̃, z̃).
        If α̃ < U − ε then
            choose a branching variable according to one of the following cases:
              • A binary variable z_i with z̃_i ∉ {0, 1}.
              • A variable p_u or q_a in a δ1-violated constraint G(p̃, q̃, z̃) ≤ 0.
              • A variable p_u or q_a in the objective if C(p̃, q̃, z̃) < α̃ − ε.
              • The flow variable q_a if Algorithm 4.1 suggested to branch w.r.t. q_a = 0.
              • A variable p_u, p_v, or q_a if Algorithm 4.1 suggested to branch w.r.t. the
                "most violated" pipe a = (u, v).
            Branch w.r.t. the chosen variable and add nodes to L.
If U < ∞ then
    return (ε, δ1 + δ2)-optimal solution (p*, q*, z*)
else
    return "infeasible".
```

assume for ease of notation that $0 \leq \nu_c A\, \underline{p}_{out}^k - c\, \underline{q}_a^k$ and $0 \leq \nu_c A\, \overline{p}_{out}^k - c\, \overline{q}_a^k$ is true for all nodes $\mathcal{F}_k$. Note that in practice this can be assured by bound propagation. Then due to monotonicity, the inequality

$$P_a^u\big(\underline{p}_{out}^k, \underline{q}_a^k, N_a\big) \leq P_a^u\big(p_{out}, q_a, N_a\big) \leq P_a^u\big(\overline{p}_v^k, \overline{q}_a^k, N_a\big)$$

is fulfilled for all nodes k and all $(p_{out}, q_a) \in U_a^k$. Therefore, the concave envelope $\operatorname{cave}_{U_a^k}[P_a^u]$ of P_a^u satisfies the inequality $\operatorname{cave}_{U_a^k}[P_a^u](p_{out}, q_a) \leq P_a^u(\overline{p}_v^k, \overline{q}_a^k, N_a)$. Hence, if

$$P_a^u\big(\overline{p}_v^k, \overline{q}_a^k, N_a\big) - P_a^u\big(\underline{p}_v^k, \underline{q}_a^k, N_a\big) \leq \delta_1 \tag{4.23}$$

holds, δ_1-infeasible points can be cut off by the concave envelope, i.e., $(p_{in}^k, p_{out}^k, q_a^k)$ is δ_1-feasible.

By construction of P_a^u it is continuous and converges to the solution of the ODE for $N_a \to \infty$, therefore N_a is only increased a finite number of times. Thus, by the continuity of P_a^u and condition (3.7), i.e., $\lim_{k\to\infty} \operatorname{diam} \mathcal{F}_k = 0$, we can derive that an index $k_0 \in \mathbb{N}$ exists such that inequality (4.23) holds for all $k \geq k_0$. Therefore, condition (3.11) holds for all $k \geq k_0$. □

Proposition 4.5 and Proposition 4.6 show that our construction of under- and overestimators for the Euler equation satisfies the necessary requirements of Theorem 3.8.

Corollary 4.7. *Suppose that conditions* (3.7) *and* (3.10) *hold. Then for* $\varepsilon > 0$, $\delta_1 > 0$, $\delta_2 > 0$, $\nu_c \in (0, 1)$ *and* N_0 *chosen appropriately Algorithm 4.2 terminates after a finite number of iterations with an* $(\varepsilon, \delta_1 + \delta_2)$*-optimal solution of* (4.18) *or the conclusion that the problem is infeasible.*

This corollary shows that our approach and Algorithm 4.2 works for the example of stationary gas transport.

4.5 Possible Extensions to the Model

In this section, we discuss two changes to our model and how we can handle these in the Algorithms 4.1 and 4.2. In Section 1.1, and Section 4.2 we assumed that the friction coefficient λ is constant and that the pipes are horizontal, i.e., the slope satisfies $\sigma = 0$. However, for the friction coefficient there are several flow dependent approximations and formulas other than the formula of Nikuradse [106, 107] known in the literature; for example, see Koch et al. [82] and the references therein. Moreover, in the real world of course not all pipes are horizontal. In the following, we show how

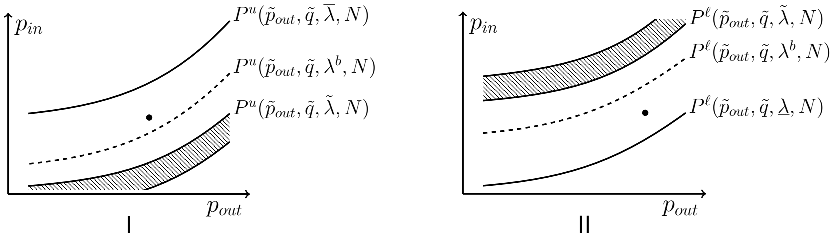


Figure 4.6. Resolving infeasibility for a solution $(\tilde{p}_{in}, \tilde{p}_{out}, \tilde{q}, \tilde{\lambda})$ of the relaxation is more involved with nonconstant friction coefficient, since under- and overestimators have to be constructed w.r.t. $\underline{\lambda}$ and $\overline{\lambda}$. The left-hand side depicts Case I, where $\tilde{p}_{in}$ is too large, and the right-hand side depicts Case II with $\tilde{p}_{in}$ too small.

to reflect the corresponding changes to the model in our spatial branch-and-bound approach. However, note that these changes have not been implemented.

4.5.1 Nonconstant Friction Coefficient

The formula of *Hagen-Poisseuille* [37], the equation of *Hofer* [71] and the equation of *Prandtl and Colebrook* [25] are alternatives to using the formula of Nikuradse (1.3). In particular, the equation of Prandtl and Colebrook is considered to be the most accurate approximation for the friction coefficient. In these models, the friction coefficient λ depends on the mass flow q.

To include a variable friction coefficient in our model, we assume that the friction coefficient is bounded by $\underline{\lambda} \le \lambda \le \overline{\lambda}$ and add the corresponding formula/equation to the constraints. We remark that the friction coefficients named above can be treated by standard MINLP techniques.

By choosing the number of discretization steps sufficiently big, we can still use P^u and P^ℓ to compute lower and upper bounds. If we interpret these also as functions in λ, then we can show that both are nondecreasing w.r.t. λ. Unfortunately though, φ, P^u and P^ℓ are not convex in (p, q, λ). Hence, we cannot simply extend the lower and upper bounding procedure of Algorithm 4.1. However, it is still possible to cut off infeasible solutions by combining branching w.r.t. λ and the cut off procedure used in Algorithm 4.1 as follows.

Using the monotonicity of P^u and P^ℓ, we can derive the inequality

$$P^\ell(p_{out}, q, \underline{\lambda}, N) \le P^\ell(p_{out}, q, \lambda, N) \le p_{in} \le P^\ell(p_{out}, q, \lambda, N) \le P^u(p_{out}, q, \overline{\lambda}, N)$$

for all feasible $(p_{in}, p_{out}, q, \lambda)$ with $q \geq 0$. Since P^u and P^ℓ are still convex w.r.t. (p, q), this shows that we can construct under- and overestimators as before, but now w.r.t. the minimal and maximal friction coefficient, respectively. However, it is now more complicated to cut off infeasible solutions. For example, consider a point $(\tilde{p}_{in}, \tilde{p}_{out}, \tilde{q}, \tilde{\lambda})$ with

$$P^\ell(\tilde{p}_{out}, \tilde{q}, \tilde{\lambda}, N) > \tilde{p}_{in} \geq P^\ell(\tilde{p}_{out}, \tilde{q}, \underline{\lambda}, N).$$

In this case, we cannot separate the solution by adding a gradient cut w.r.t. $(\tilde{p}_{out}, \tilde{q})$. Even if we perform branching w.r.t. $\lambda = \tilde{\lambda}$, then we can separate the solution in the node corresponding to the variable bounds $[\tilde{\lambda}, \overline{\lambda}]$, but not in the the node given by the variable bounds $[\underline{\lambda}, \tilde{\lambda}]$. Similar problems arise for solutions $(\tilde{p}_{in}, \tilde{p}_{out}, \tilde{q}, \tilde{\lambda})$ with

$$P^u(\tilde{p}_{out}, \tilde{q}, \tilde{\lambda}, N) < \tilde{p}_{in} \leq P^u(\tilde{p}_{out}, \tilde{q}, \overline{\lambda}, N).$$

We can solve this problem by including an additional branching step in Algorithm 4.1. Therefore, we consider the cases depicted in Figure 4.6, i.e., the cases $\tilde{p}_{in} > P^u(\tilde{p}_{out}, \tilde{q}, \tilde{\lambda}, N)$ and $P^\ell(\tilde{p}_{out}, \tilde{q}, \tilde{\lambda}, N) > \tilde{p}_{in}$. Then infeasibility can be resolved as follows.

Case I We choose λ^b with $\overline{\lambda} > \lambda^b > \tilde{\lambda}$ and $\tilde{p}_{in} > P^u(\tilde{p}_{out}, \tilde{q}, \lambda^b, N)$. After branching w.r.t. $\lambda = \lambda^b$, the current solution is not feasible in the node corresponding to variable bounds $[\lambda^b, \overline{\lambda}]$ any more. Furthermore, in the second node given by $[\underline{\lambda}, \lambda^b]$, we can proceed as in Algorithm 4.1, i.e., either add the concave envelope of P^u w.r.t. λ^b or perform spatial-branching w.r.t. p_{in}, p_{out} or q and then add a cut to separate the infeasible solution $(\tilde{p}_{in}, \tilde{p}_{out}, \tilde{q}, \tilde{\lambda})$.

Case II We choose λ^b with $\tilde{\lambda} > \lambda^b > \underline{\lambda}$ and $P^u(\tilde{p}_{out}, \tilde{q}, \lambda^b, N) > \tilde{p}_{in}$. After branching w.r.t. $\lambda = \lambda^b$, we can separate the solution in the node $[\lambda^b, \overline{\lambda}]$ by adding a gradient cut. Moreover, in the node $[\underline{\lambda}, \lambda^b]$, the solution is not feasible any more.

By integrating these two branching steps w.r.t. λ directly before Line 15 in Algorithm 4.1, we can then also handle models with nonconstant friction coefficient.

4.5.2 Pipelines with Height Differences

Recall from Section 2.4 that we studied the application of the second order Taylor method (2.12) and the trapezoidal rule (2.14) to the stationary isothermal Euler equation with height differences

$$\partial_x p(x) = \varphi_\sigma\big(p(x), q\big), \quad x \in [0, L], \quad \varphi_\sigma(p, q) := -\frac{1}{2}\frac{p}{c^2 D}\frac{2Dg\sigma A^2 p^2 + \lambda c^4 q^2}{A^2 p^2 - c^2 q^2}.$$

Like before, we denote with $P^{ta}, P^{tr} : U \times \mathbb{N} \to \mathbb{R}$ the functions defined through evaluating (2.22) and (2.23). We note that we do not use P^u and P^ℓ here, since not always the same scheme defines the lower or upper bound.

Assuming $\lambda c^2 \geq 6\, D\, g|\sigma|$, we can choose the number of discretization steps N such that the conditions of Lemmas 2.26 to 2.28 are satisfied. We remark that this assumption is not very restrictive, because we typically have $c^2 \gg 6\, D\, g|\sigma|$. Then for a pipe with $q \geq 0$ the functions P^{ta} and P^{tr} define lower and upper bounds on $p(0)$ in the following cases:

1. If $\sigma > 0$, then P^{ta} defines a convex lower bound and P^{tr} defines a convex upper bound.
2. If $\sigma < 0$ and $p(L) \leq p_r(q, \sigma)$ or $\sigma = 0$, then P^{ta} defines a convex lower bound and P^{tr} defines a convex upper bound.
3. If $\sigma < 0$ and $p(L) \geq p_r(q, \sigma)$, then P^{ta} defines a convex upper bound and P^{tr} defines a convex lower bound.

That is, in the first two cases we have

$$P^{ta}\big(p(L), q, N\big) \leq p(0) \leq P^{tr}\big(p(L), q, N\big),$$

while in the third case the roles of the Taylor method and the trapezoidal rule are reversed, i.e.,

$$P^{tr}\big(p(L), q, N\big) \leq p(0) \leq P^{ta}\big(p(L), q, N\big).$$

To handle pipelines with height differences in our model, we can construct the linear under- and overestimators similar to before in each of these cases. Therefore, we have to differentiate the three cases in Algorithm 4.1. This can be done as follows. If, after fixing the flow direction, either $q \geq 0$ and $\sigma \geq 0$ or $q \leq 0$ and $\sigma \leq 0$ holds, then we are in case 1. Otherwise, we are either in case 2 or 3. If either $\overline{p}_{out} \leq p_r(\underline{q}, \sigma)$ or $\underline{p}_{out} \geq p_r(\overline{q}, \sigma)$ holds, then either case two or three applies. If neither the lower or upper pressure bound satisfies these conditions, then we have to perform an additional branching step before we can compute under- and overestimators. Note that unlike before it does not suffice to perform branching w.r.t. a fixed pressure value. Instead we have to perform so-called constraint branching, i.e., we create two new nodes and add the constraints

$$p_{out} \leq p_r(q, \sigma) = \left(\frac{c^2}{A} \sqrt{\frac{-\lambda}{2\, D\, g\, \sigma}} \right) q,$$

and

$$p_{out} \geq p_r(q, \sigma) = \left(\frac{c^2}{A} \sqrt{\frac{-\lambda}{2\, D\, g\, \sigma}} \right) q,$$

respectively. Afterwards, we can construct the linear under- and overestimators. However, in case 2 and 3 we have to take the additional constraints into account when constructing the concave envelope of the upper bound. That is, we have to consider other domains U than those depicted in Figure 4.3. Moreover, to add gradient cuts in case 3 we have to compute the derivatives $\partial_p P^{tr}$ and $\partial_q P^{tr}$ of the lower bound, which is given by evaluating the implicit trapezoidal rule. To avoid this, we can consider an initial value problem instead of the end value problem (2.21) in case 3 and compute bounds on the output pressure; see also Section 2.4 and Remark 2.29.

4.6 First Numerical Results

We have implemented the models presented in Section 4.2 and Algorithm 4.2 using Algorithm 4.1 to construct the relaxations for the ODE constraints (4.4) with the branch-and-bound framework SCIP [40, 132] using CPLEX as LP-solver. At this point we refer to Chapter 6 for details on the implementation and the computational setup.

Among other instances, we use instance `GasLib-40` from `GasLib` [41, 126], which is a library of gas network instances. The network has 40 nodes, 39 pipes, and 6 compressor stations. There is one load scenario, that is, in- and outflows, available online at [41], which has 3 entries and 29 exits. With the first running version of our implementation we were able to solve problem (4.18) with the objective to maximize the sum of pressures within one hour. On the one hand, this proved that our approach also works in practice, on the other hand, the running times were not satisfactory. Moreover, we were not able to solve any bigger instances.

Note that although we could not recreate similar running times with the current version of our implementation, the following results have been achieved with the current version of our code. A partial explanation for the faster running times is that we now use SCIP version `7.0.0` and CPLEX version `12.10.0` instead of version `3.2.1` and version `12.6.1`, respectively, and of course there have been major improvements of the code. Nevertheless, using the basic algorithmics presented in this chapter and, in particular, not using the flow tightening techniques which will be discussed in Section 6.2, the solving process with the current version of our implementation shows similar characteristics.

We solved instance `GasLib-40` with $\nu_c = 0.8$ and with feasibility tolerances 10^{-6} (SCIP default value) for solving the LP-relaxations and $\delta_1 = \delta_2 = 10^{-4}$ for the ODE constraints, see Corollary 4.7. The computation took 972.90 seconds and 24 138 nodes were processed in the branch-and-bound tree. Analyzing the solving process, we observe that the flow variables of 13 pipes were already fixed after presolving. However, Figure 4.7 shows that, except for one pipe, these pipes are at tree-like ends of the network. For the remaining 26 pipelines the mean lower and upper

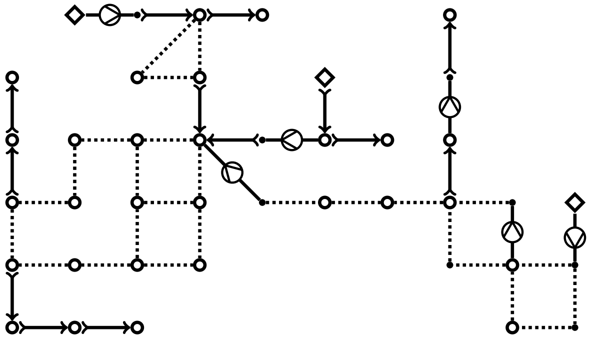


Figure 4.7. The network `GasLib-40` after presolving for the scenario which has 3 sources (diamonds) and 29 sinks (circles). Pipes with fixed flow are depicted by $\rightarrowtail$, the remaining pipes are dashed.

bounds of the flow variables are $-2125.83\,\mathrm{kg/s}$ and $2139.25\,\mathrm{kg/s}$, which is almost no improvement to the initial bounds of $\pm 2180.56\,\mathrm{kg/s}$, and nowhere near to the mean absolute flow of $62.63\,\mathrm{kg/s}$ in the optimal solution. In fact, Figure 4.7 also shows that even for some bridges in the graph the flow could not be fixed during presolving, even though it is uniquely defined by flow conservation.

As the main reason for the long running time and the poor performance of the presolving we identified the fact that the flow direction has to be fixed before we can construct the LP-relaxation for the gas flow; see Section 4.3. Hence, since the lower and upper flow bounds provided in the description of the network `GasLib-40` are negative, and positive, respectively, we use the variable bounds as initial relaxation. Thus, the relaxation does not represent the simplest physical properties of the gas flow, e.g., that the pressure decreases in the direction of the flow.

As a consequence of this, we implemented problem specific bound tightening for the flow variables; see Section 6.2. Furthermore, consider a pipe $a = (u, v) \in \mathcal{A}^{pi}$. To couple the pressure difference $p_u - p_v$ with the flow direction, we introduce binary variables z_a^+ and z_a^- to model the flow direction. The binary variables are coupled with the flow q_a through the constraints

$$z_a^+ + z_a^- \leq 1 \qquad \text{and} \qquad \underline{q}_a\, z_a^- \leq q_a \leq \overline{q}_a\, z_a^+.$$

Then, if the flow is positive, $z_a^+ = 1$ holds and if the flow is negative, $z_a^- = 1$ holds. Note that we use two binary variables z_a^+ and z_a^- with $z_a^+ + z_a^- \leq 1$ instead of a single binary variable z with $\underline{q}_a\,(1 - z) \leq q_a \leq \overline{q}_a\, z$ such that we do not have to assign a direction to flow $q_a = 0$, see also the next chapter. Hence, by including the inequality

$$(\underline{p}_u - \overline{p}_v)\, z_a^- \leq p_u - p_v \leq (\overline{p}_u - \underline{p}_v)\, z_a^+ \tag{4.24}$$

in our model we can represent the physical property that the pressure is nonincreasing in the direction of the flow and thus strengthen the relaxation. Furthermore, note that an immediate implication of this property is that gas cannot flow in cycles unless the pressure is increased via a compressor station. To further strengthen our model, this observation motivated the study of acyclic flows in the subsequent chapter.

CHAPTER 5

Combinatorial Models for Acyclic Flows

In the previous chapter, we applied the spatial branch-and-bound algorithm which was developed in Chapter 3 to stationary gas transport and discussed first numerical results in Section 4.6. There we noticed that stationary gas flow is necessarily acyclic (unless the pressure is increased in compressor stations). Since the numerical results and, in particular, the solving times were not convincing, this motivated us to study acyclic flows.

The observation that the flow is acyclic does not only apply to stationary gas transport, but also to potential-based flows, which form a basic model for physical networks. Note that when using the Weymouth equation (1.9) to describe gas flow in pipelines, stationary gas transport is a special case of potential-based flows. Moreover, other networks such as water networks and DC power flow networks can be formulated as potential-based flows. Thus, we study acyclic potential-based flows in this chapter, instead of directly applying the theory to gas transport.

To introduce the fundamental ideas of this chapter and to show that potential-based flows are necessarily acyclic, we first present the basic setting of potential-based flows. Note that we will provide references to corresponding literature later on. Let $\mathcal{D} = (\mathcal{V}, \mathcal{A})$ be a simple directed graph without anti-parallel arcs. We assume that for all arcs $a \in \mathcal{A}$, there is a continuous, strictly increasing *potential function* $\psi_a \colon \mathbb{R} \to \mathbb{R}$ with $\psi_a(0) = 0$ as well as a *resistance* $\beta_a > 0$. Each node $v \in \mathcal{V}$ has an associated *potential* π_v. Under mild assumptions on the potential functions and given demand on the nodes, the defining equations

$$\pi_u - \pi_v = \beta_a \, \psi_a(x_a) \quad \forall \, a = (u, v) \in \mathcal{A}, \tag{5.1}$$

induce a unique flow $x \in \mathbb{R}^{\mathcal{A}}$ and unique potential differences. For physical networks, the potentials π_u correspond to quantities like squared pressure or voltage. Note that a directed instead of undirected graph $\mathcal{D}$ is used to define a direction of the flow and (5.1). Thus, flow values can also be negative, indicating flow in the opposite direction of the arc.

A flow $x \in \mathbb{R}^{\mathcal{A}}$ defines a directed graph $\mathcal{D}(x) := (\mathcal{V}, \mathcal{A}(x))$ with

$$\begin{aligned}\mathcal{A}(x) := &\big\{(u,v) \in \mathcal{V} \times \mathcal{V} \,:\, (u,v) \in \mathcal{A} \text{ with } x_{(u,v)} > 0\big\} \\ \cup &\big\{(v,u) \in \mathcal{V} \times \mathcal{V} \,:\, (u,v) \in \mathcal{A} \text{ with } x_{(u,v)} < 0\big\}.\end{aligned}$$

If x satisfies (5.1), we claim that this graph is always acyclic. To see this, assume that there would exist a directed cycle $C \subseteq \mathcal{A}(x)$. Then by splitting the arcs in C into forward arcs, i.e., those contained in the original graph, and backward arcs, i.e., those which have the opposite direction to the original graph, we obtain

$$\begin{aligned}&\sum_{(u,v)\in C\cap\mathcal{A}} \beta_{(u,v)}\, \psi_{(u,v)}(x_{(u,v)}) - \sum_{(u,v)\in C\setminus\mathcal{A}} \beta_{(v,u)}\, \psi_{(v,u)}(x_{(v,u)}) \\ &\qquad = \sum_{(u,v)\in C\cap\mathcal{A}} \big(\pi_u - \pi_v\big) + \sum_{(u,v)\in C\setminus\mathcal{A}} \big(\pi_u - \pi_v\big) = 0,\end{aligned}$$

where we use that C is a cycle and thus the alternating sum of the potentials vanishes. However, since the resistances β_a are positive and each ψ_a is strictly increasing with $\psi_a(0) = 0$, we have $\beta_a\, \psi_a(x_a) > 0$ if $x_a > 0$ and $\beta_a\, \psi_a(x_a) < 0$ if $x_a < 0$. Thus, the value of the first line is positive, leading to a contradiction. This shows that $\mathcal{D}(x)$ is acyclic, corresponding to the physical property of having a conservative potential.

Uniqueness and acyclicity are two important physical properties that are captured by potential-based flows. Moreover, this model class is important for handling energy networks other than gas networks; see Section 5.2 for examples. Such networks often contain *active* network elements like switches/valves, allowing to close a connection, or generators/pumps/compressors, which can increase the potential on certain arcs. These elements allow to control flow and potentials. Their presence may violate acyclicity, which provides more freedom to control the network. However, the passive components remaining after removal of active elements satisfy acyclicity.

Using the degrees of freedom of active elements, several different optimization problems over such networks are interesting, e.g., energy minimal operation under the assumption that a given flow demand is satisfied. When solving such optimization problems to global optimality, one can exploit the fact that the passive components of the network still have an acyclic flow. We will demonstrate how to enhance existing mixed-integer nonlinear programing formulations using binary variables for the flow directions and constraints that enforce acyclicity. Therefore, we will use the example of stationary gas transport introduced in the previous chapter. Yet, we use the Weymouth equation (1.9) instead of the ordinary differential equations (4.4) to describe gas flow in pipes.

This chapter is structured as follows. After a short literature review we complete the setting of potential-based flows and provide examples in Section 5.2. Section 5.3 first introduces a nested sequence of polytopes which provide combinatorial models for the directions of the flows. Thereby we relax more and more constraints of the nonlinear model of potential-based flows along the way. Their relation is studied in Section 5.3.1. In Section 5.3.2 we investigate a model solely based on acyclic directions and the complexity of optimizing over the corresponding polytope. Section 5.3.3 then introduces our main model, exploiting both acyclicity and the fact that one needs to connect sources and sinks. To see that this model provides a good compromise between the nonlinear model and the so-called acyclic subgraph polytope, the particular model is investigated in Sections 5.3.4 and 5.3.5 in more detail. Then in Section 5.4, we add the corresponding inequalities to problem (4.18) using the potential-based flow model instead of the ODE model. We demonstrate that this approach leads to an improvement of the solving time by about a factor of 3 on average and a significant speed-up for the time to prove optimality by almost a factor of 5. Furthermore, in Chapter 6 we will use those inequalities to speed-up our spatial branch-and-bound algorithm for the model with the ODE constraints.

We remark that the contents of this chapter are available in similar form online in the article [55] which is joint work with Marc E. Pfetsch. Furthermore, this article is submitted for publication in an international journal. The presentation of the numerical results in Section 5.4 is adjusted to this thesis and somewhat extended.

5.1 Literature Review

Potential-based flows have been studied repeatedly in the literature and have been used in many different contexts. Hendrickson and Janson [67] provide an overview. It seems that the first appearance is in Birkhoff and Diaz [13]. A general treatment appears in Rockafellar [117]. We will refer to more literature in Section 5.2 and also discuss examples there. The special case of stationary gas transport, which was already introduced in Chapter 4, will be used in our computations in Section 5.4.

The topic of acyclic flows for potential-based flow has been investigated by Becker and Hiller in four articles [7, 8, 9, 68]. Their motivation is similar to ours and they also test their methods on gas networks. The main combinatorial model of these articles is based on so-called acyclic source transshipment sink (ASTS) orientations. We will arrive at an equivalent definition through a polyhedral approach in Section 5.3.3. Their contributions can be briefly summarized as follows. A characterization of the cases in which an ASTS orientation exists is given in [8]. Moreover, various decomposition results based on 2-connected components of the underlying graph are presented in [7, 9]. This allows to preprocess the networks [7, Section 3]. The enumeration of ASTS orientations is discussed in [7, 9]. This is used in a Dantzig-Wolfe type approach to strengthen potential-based flow formulations for gas networks in [7, 9].

Our results differ from the ones by Becker and Hiller in the following way. We embed the common combinatorial model in a sequence of polytopes and we investigate their properties. We use a different binary encoding, provide a different setup, investigate complexity results, and add the inequalities to the model instead of using enumeration of the possible orientations.

The ideas of this chapter were already used for the computations in [56] and we provide the background here. Apart from the mentioned literature, we are not aware of any other works concerning combinatorial models for acyclic flows.

5.2 Potential-based Flows

Before referring to more results in the literature and starting our study of acyclic flows in the next section, we first complete the setting of potential-based flows.

Recall that we assume that $\mathcal{A}$ contains at most one of (u, v) and (v, u) for every pair of nodes u, $v \in \mathcal{V}$. For a subset $U \subseteq \mathcal{V}$ we use the shorthand

$$\delta^+(U) := \{(u, w) \in \mathcal{A} \,:\, u \in U,\ w \notin U\}$$

for the outgoing arcs and analogously

$$\delta^-(U) := \{(w, u) \in \mathcal{A} \,:\, u \in U,\ w \notin U\}$$

for the ingoing arcs. We use the abbreviation $\delta^+(v) := \delta^+(\{v\})$ and $\delta^-(v) := \delta^-(\{v\})$ for $v \in \mathcal{V}$.

For every node $v \in \mathcal{V}$, there are lower and upper potential bounds $\underline{\pi}_v$ and $\overline{\pi}_v \in \mathbb{R}$, respectively, with $\underline{\pi}_v \leq \overline{\pi}_v$. Additionally, for every arc $a \in \mathcal{A}$, there are lower and upper flow bounds $\underline{x}_a$ and $\overline{x}_a \in \mathbb{R}$, respectively, with $\underline{x}_a \leq \overline{x}_a$. Let $b \in \mathbb{R}^{\mathcal{V}}$ be a supply and demand vector that is *balanced*, i.e., $\sum_{v \in \mathcal{V}} b_v = 0$. A node $v \in \mathcal{V}$ is called *source node* if $b_v > 0$, *sink node* if $b_v < 0$, and *inner node* if $b_v = 0$. Then $(x, \pi) \in \mathbb{R}^{\mathcal{A}} \times \mathbb{R}^{\mathcal{V}}$ is a (passive) *potential-based flow* if it satisfies the following constraints:

$$\sum_{a\in\delta^+(v)} x_a - \sum_{a\in\delta^-(v)} x_a = b_v \qquad \forall\, v \in \mathcal{V}, \tag{5.2a}$$

$$\pi_u - \pi_v = \beta_a\, \psi_a(x_a) \qquad \forall\, a = (u,v) \in \mathcal{A}, \tag{5.2b}$$

$$\underline{\pi}_v \le \pi_v \le \overline{\pi}_v \qquad \forall\, v \in \mathcal{V}, \tag{5.2c}$$

$$\underline{x}_a \le x_a \le \overline{x}_a \qquad \forall\, a \in \mathcal{A}. \tag{5.2d}$$

We call $x \in \mathbb{R}^{\mathcal{A}}$ a *b-flow* if it satisfies (5.2a).

Throughout this chapter, we will assume that each potential function $\psi_a \colon \mathbb{R} \to \mathbb{R}$, for $a \in \mathcal{A}$, is continuous and strictly increasing with $\psi_a(0) = 0$. For some results in the literature, additional requirements needed, e.g., that the potential functions are odd (i.e., $\psi_a(x) = -\psi_a(-x)$), positively homogeneous (i.e., $\psi_a(\lambda x) = \lambda^r \psi_a(x)$ for all $x \in \mathbb{R}$ and $\lambda > 0$ with some constant $r > 0$) or that they are the same for every arc, see, e.g., Groß et al. [49], but we do not need these assumptions here.

One important result, see Maugis [98], Collins et al. [26], and Ríos-Mercado et al. [115] is the following: Assume that $\mathcal{D}$ is weakly connected, there are no bounds on the potentials and flows, for a given node $s \in \mathcal{V}$ the potential π_s is fixed, and the potential functions are continuous and strictly increasing. Then there exists a unique feasible potential-based flow (x, π). Consequently, System (5.2) with a fixed potential π_s is either infeasible or has a unique solution. One tool for proving uniqueness is a cost minimal flow problem with a strongly convex objective, whose dual multipliers provide the potentials, see Maugis [98], Rockafellar [117], and Groß et al. [49] for more information and a discussion of the corresponding Lagrange dual.

Example 5.1. Several interesting applications can be modeled as potential-based flows, see, e.g., Hendrickson and Janson [67]. We present three energy network examples:

1. Stationary Gas Transport Networks: Here arcs correspond to pipelines, the potentials are the squares of pressures, and flows are gas mass flows. One common approximation of gas flow is (5.1) with $\psi_a(x_a) = |x_a|\, x_a$, which is also called the Weymouth equation (1.9). The positive arc-specific constant β_a depends on the pipelines diameter, length, and roughness of its inner wall. More details on stationary gas flow in pipeline networks are given in the previous chapter, in the book by Koch et al. [82], and the references therein. The above model assumes constant heights of the network, but one can use scaling to incorporate different heights, see Groß et al. [49].

   The computational results in Section 5.4 are based on gas networks, extended by active elements like valves and compressors.

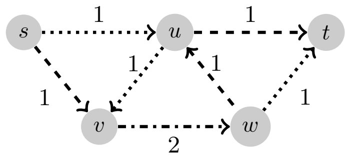


Figure 5.1. An s–t-flow which can be decomposed into two paths (indicated by dashed/dotted arcs) each with flow value 1, but that is not acyclic.

2. Water Networks: Here potentials correspond to hydraulic heads. Common potential functions are $\psi_a(x_a) = \operatorname{sgn}(x_a)\,|x_a|^{1.852}$, see, e.g., Larock et al. [86], or $\psi_a(x_a) = |x_a|\,x_a$, see, e.g., Burgschweiger et al. [18].
3. Lossless DC Power Flow Networks: In this case, the potentials are voltages and the potential function is linear $\psi_a(x_a) = x_a$. For more information about power flow network planning, we refer to Bienstock [12].

5.3 Combinatorial Models for Acyclic Flows

To study acyclic flows, we first introduce some basic notation. Consider a simple and weakly connected directed graph $\mathcal{D}$ without anti-parallel arcs. Note that these assumptions are for notational convenience: Loops always have a zero flow and can be removed, anti-parallel arcs can be reoriented to be parallel. Moreover, application to energy networks motivates to use the same positively homogeneous potential function for all arcs. Then parallel arcs can be merged into one arc with adapted β-value; see Groß et al. [49]. Furthermore, each weakly connected component can be treated separately.

We will use $\overleftarrow{a}$ to denote the reverse arc of some arc $a \in \mathcal{A}$, i.e., if $a = (u, v)$, then $\overleftarrow{a} := (v, u)$. Furthermore, let $\overleftarrow{\mathcal{A}} := \{\overleftarrow{a} : a \in \mathcal{A}\}$ be the set of the reversed arcs. Note that for a given flow x, we can also write

$$\mathcal{A}(x) = \{a \in \mathcal{A} : x_a > 0\} \cup \{\overleftarrow{a} \in \overleftarrow{\mathcal{A}} : x_a < 0\},$$

and the digraph $\mathcal{D}(x) = (\mathcal{V}, \mathcal{A}(x))$ is a reorientation of the subgraph consisting of arcs with nonzero flow such that all arcs point in the direction of the flow. We say that x is an *acyclic flow* if and only if $\mathcal{D}(x)$ is acyclic. An alternative definition for acyclic flows is given by Hiller and Becker [68, Definition 1].

Remark 5.2. Note that being an acyclic s–t-flow is not related to having a flow decomposition which only consists of paths and no cycles. On the one hand, the

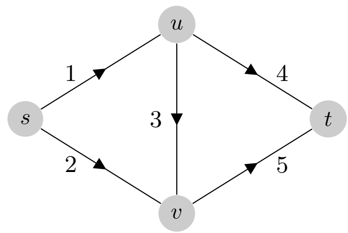


Figure 5.2. A diamond shaped network with source s and sink t and arc labels 1 to 5.

sum of flow along paths can contain a cycle. For example, consider the flow network given in Figure 5.1. The s–t-flow depicted in this figure can be decomposed into the flows x^1 (dashed) and x^2 (dotted). Both x^1 and x^2 are flows along paths and therefore acyclic, nevertheless their sum is not, since $\mathcal{D}(x^1+x^2)$ contains the directed cycle $C = (u, v, w, u)$. On the other hand, the sum of flows along paths and cycles can be acyclic. For example consider flow x^3 with value -1 along C. Then, the digraph $\mathcal{D}(x^1 + x^2 + x^3)$ is acyclic.

The following example shows how flow directions can depend on the resistances β_a.

Example 5.3. Consider the potential network given in Figure 5.2 with source s and sink t and $b_s = -b_t > 0$. Let $b_u = b_v = 0$.

By constraint (5.2a), it is clear that $x_1+x_2 = x_4+x_5 = b_s > 0$ has to hold for every potential-based flow (x, π). Furthermore, x_1, x_2, x_4 and x_5 have to be nonnegative because x is necessarily acyclic: Assume that one of them is negative. We can assume w.l.o.g. $x_2 < 0$ by symmetry. Then $x_1 > 0$ holds by flow conservation. Having $x_3 > 0$ would close the cycle (s, u, v, s). Thus, $x_3 \leq 0$ and by flow conservation $x_5 < 0$. Since also $x_4 > 0$, this closes the cycle (s, u, t, v, s).

Thus, except for arc (u, v), all flow directions for this potential network are fixed, independent of the potential functions and the β-values. However, the flow direction of (u, v) depends on the potential functions and the β-values. Indeed, even if we use the same potential function $\psi = \psi_a$ for all arcs a, the flow direction is not uniquely determined: If all β-values are equal, then the corresponding flow is $x_3 = 0$. Moreover, consider the case $\beta_1 > \beta_2 = \cdots = \beta_5$ and assume $x_3 \geq 0$. Then $\pi_u \geq \pi_v$ holds by (5.2b). Further, $\beta_1 > \beta_2$ implies $x_2 > x_1$. Due to flow conservation also $x_5 > x_4$. This gives

$$\pi_t = \pi_u - \beta_4\, \psi(x_4) > \pi_v - \beta_5\, \psi(x_5) = \pi_t,$$

which is a contradiction. Hence, $x_3 < 0$. Furthermore, by symmetry using β-values $\beta_1 < \beta_2 = \cdots = \beta_5$ implies $x_3 > 0$.

To express acyclicity, for each arc $a \in \mathcal{A}$, we introduce binary variables z_a^+ and z_a^- that model the flow direction as follows:

$$\operatorname{sgn}(x_a) = z_a^+ - z_a^-, \quad z_a^+ + z_a^- \leq 1, \tag{5.3}$$

where $\operatorname{sgn}(x_a) = -1$ if $x_a < 0$, $\operatorname{sgn}(x_a) = 0$ if $x_a = 0$, and $\operatorname{sgn}(x_a) = 1$ if $x_a > 0$. Thus, these constraints imply that $z_a^+ = 1$ holds if $x_a > 0$, $z_a^- = 1$ if $x_a < 0$, and $z_a^+ = z_a^- = 0$ if $x_a = 0$.

The total model for potential-based flows is the following:

$$\begin{aligned}
\sum_{a \in \delta^+(v)} x_a - \sum_{a \in \delta^-(v)} x_a &= b_v && \forall\, v \in \mathcal{V}, \\
\pi_u - \pi_v &= \beta_a\, \psi_a(x_a) && \forall\, a = (u,v) \in \mathcal{A}, \\
\underline{\pi}_v \leq \pi_v &\leq \overline{\pi}_v && \forall\, v \in \mathcal{V}, \\
\underline{x}_a \leq x_a &\leq \overline{x}_a && \forall\, a \in \mathcal{A}, \\
z_a^+ - z_a^- &= \operatorname{sgn}(x_a) && \forall\, a \in \mathcal{A}, \\
z_a^+ + z_a^- &\leq 1 && \forall\, a \in \mathcal{A}, \\
z_a^+,\ z_a^- &\in \{0,1\} && \forall\, a \in \mathcal{A}.
\end{aligned} \tag{5.4}$$

We define the feasible set of potential-based flows with the corresponding flow directions

$$\mathcal{X} \coloneqq \{(x, \pi, z^+, z^-) \text{ feasible for } (5.4)\} \subset \mathbb{R}^{\mathcal{A}} \times \mathbb{R}^{\mathcal{V}} \times \{0,1\}^{2\mathcal{A}}.$$

Moreover, for $(z^+, z^-) \in \{0,1\}^{2\mathcal{A}}$ we define

$$\mathcal{A}(z^+, z^-) \coloneqq \{a \in \mathcal{A} \,:\, z_a^+ = 1\} \cup \{\overleftarrow{a} \in \overleftarrow{\mathcal{A}} \,:\, z_a^- = 1\}$$

and the corresponding subgraph $\mathcal{D}(z^+, z^-) \coloneqq \big(\mathcal{V}, \mathcal{A}(z^+, z^-)\big)$ of $\overleftrightarrow{\mathcal{D}} \coloneqq (\mathcal{V}, \overleftrightarrow{\mathcal{A}})$, where we have $\overleftrightarrow{\mathcal{A}} \coloneqq \mathcal{A} \cup \overleftarrow{\mathcal{A}}$. Then we call (z^+, z^-) *acyclic* if and only if $\mathcal{D}(z^+, z^-)$ is acyclic in the directed sense.

In order to exploit acyclicity of potential-based flows, we will investigate purely combinatorial models of acyclicity, i.e., polytopes that are only based on the variables z_a^+ and z_a^-. There are several possibilities for such models, depending on how many properties of the potential-based flow are used. We present four polytopes in the following and one model in Section 5.3.3. The main goal is to derive inequalities that can be added to (5.4) in order to improve the computational solving performance.

Projected Potential-Based Flows The most specific model considers the projection of feasible points of (5.4) for a given network with given balanced $b \in \mathbb{R}^{\mathcal{V}}$ and yields the *polytope of potential-based flow directions*

$$\mathcal{P}^{\mathrm{PF}} := \operatorname{conv}\left\{ \begin{pmatrix} z^+ \\ z^- \end{pmatrix} : \exists\, (x, \pi) \in \mathbb{R}^{\mathcal{A}} \times \mathbb{R}^{\mathcal{V}} \text{ with } (x, \pi, z^+, z^-) \in \mathcal{X} \right\}.$$

Note that since the flows are unique, (z^+, z^-) is also unique. The only possible variation is whether $\mathcal{P}^{\mathrm{PF}}$ is empty or not. Since it is an NP-hard problem to decide whether there exists a potential-based flow for the case of DC-flows, see Lehmann et al. [88], and for the case of gas networks, see Szabó [146], it is an NP-hard problem to decide whether $\mathcal{P}^{\mathrm{PF}}$ is empty.

Projected Universal Potential-Based Flows Example 5.3 shows that the flow directions can depend on the values of β. We therefore investigate a model in which the resistances β are allowed to vary.

Consider the asymptotic behavior of $\beta_a^{-1}(\pi_u - \pi_v) = \psi_a(x_a)$ for $\beta_a \to \infty$. For fixed potentials and $\beta_a \to \infty$ we get $x_a \to 0$. Thus, we identify (5.2b) for $\beta_a = \infty$ with the constraint $x_a = 0$ and decoupled potentials π_u, π_v. This has the same effect as if arc $a = (u, v)$ would not exist. In the following we denote the extended real line with $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, \infty\}$.

Again given a digraph $\mathcal{D}$ with balanced $b \in \mathbb{R}^{\mathcal{V}}$, we define the *polytope of universal potential-based flow directions* for (5.4) as

$$\mathcal{P}^{\mathrm{UPF}} := \operatorname{conv}\left\{ \begin{pmatrix} z^+ \\ z^- \end{pmatrix} : \exists\, \beta \in \overline{\mathbb{R}}^{\mathcal{A}}_{>0},\ (x, \pi) \in \mathbb{R}^{\mathcal{A}} \times \mathbb{R}^{\mathcal{V}} \text{ with } (x, \pi, z^+, z^-) \in \mathcal{X} \right\}.$$

By allowing the resistances β to vary, $\mathcal{P}^{\mathrm{UPF}}$ abstracts from the particular network to some extent. Note that the polytope is universal in the sense that changes in β allow a corresponding change of direction of some arcs as in Example 5.3.

Acyclic Flows The *polytope of acyclic flow directions* is

$$\mathcal{P}^{\mathrm{AF}} := \operatorname{conv}\left\{ \begin{pmatrix} z^+ \\ z^- \end{pmatrix} \in \{0, 1\}^{2\mathcal{A}} : \exists\, x \in \mathbb{R}^{\mathcal{A}} \text{ s.t. } (5.2a), (5.2d), (5.3),\ \mathcal{D}(x) \text{ acyclic} \right\}.$$

That is, $\mathcal{P}^{\mathrm{AF}}$ is the convex hull of all binary vectors $(z^+, z^-) \in \{0,1\}^{2\mathcal{A}}$ such that there exists an acyclic flow $x \in \mathbb{R}^{\mathcal{A}}$, i.e., $\mathcal{D}(x)$ is acyclic, satisfying the constraints

$$\begin{aligned}
\sum_{a \in \delta^+(v)} x_a - \sum_{a \in \delta^-(v)} x_a &= b_v && \forall\, v \in \mathcal{V}, \\
\underline{x}_a \le x_a &\le \overline{x}_a && \forall\, a \in \mathcal{A}, \\
z_a^+ - z_a^- &= \operatorname{sgn}(x_a) && \forall\, a \in \mathcal{A}, \\
z_a^+ + z_a^- &\le 1 && \forall\, a \in \mathcal{A}.
\end{aligned} \tag{5.5}$$

Note that from $\mathcal{P}^{\mathrm{UPF}}$ to $\mathcal{P}^{\mathrm{AF}}$ the potential equation (5.2b) of (5.4) is relaxed. Because of (5.3), we could replace the requirement that $\mathcal{D}(x)$ is acyclic by acyclicity of $\mathcal{D}(z^+, z^-)$.

Acyclic Subgraphs The polytope that abstracts the most from potential-based flows is the *polytope of acyclic subgraphs*

$$\mathcal{P}^{\mathrm{AS}} := \operatorname{conv} \left\{ \begin{pmatrix} z^+ \\ z^- \end{pmatrix} \in \{0,1\}^{2\mathcal{A}} \,:\, \mathcal{D}(z^+, z^-) \text{ is acyclic} \right\}.$$

Note that acyclicity of $\mathcal{D}(z^+, z^-)$ implies $z_a^+ + z_a^- \le 1$ for each arc $a \in \mathcal{A}$. In Section 5.3.3, we will refine this model by using knowledge of sources, sinks, and inner nodes.

Remark 5.4. There are two alternatives to using (5.3). The first one is

$$\underline{x}_a\, z_a^- \le x_a \le \overline{x}_a\, z_a^+, \quad z_a^+ + z_a^- \le 1. \tag{5.6}$$

These constraints form a relaxation of (5.3), since the direction variables z_a^+ and z_a^- can be chosen freely if $x_a = 0$ (as long as $z_a^+ + z_a^- \le 1$). For $(x, \pi, z^+, z^-) \in \mathcal{X}$ this would allow $\mathcal{D}(z^+, z^-)$ to have cycles although $\mathcal{D}(x)$ is acyclic. Thus, the polytopes $\mathcal{P}^{\mathrm{PF}}$ and $\mathcal{P}^{\mathrm{UPF}}$ would contain (z^+, z^-) that do not correspond to potential-based flows. Note that this model is only valid if $\underline{x} \le 0 \le \overline{x}$, because positive bounds $\underline{x}$ or negative bounds $\overline{x}$ would be overruled when setting $z^+ = 0$ or $z^- = 0$.

The second alternative for (5.3) is

$$\underline{x}_a\, z_a^- \le x_a \le \overline{x}_a\, z_a^+, \quad z_a^+ + z_a^- = 1.$$

This would require to assign a direction to a 0-flow arc, which would make the following analysis more difficult.

5.3.1 Relations among Combinatorial Models

We begin with the obvious observation that

$$\mathcal{P}^{\mathrm{PF}} \subseteq \mathcal{P}^{\mathrm{UPF}} \subseteq \mathcal{P}^{\mathrm{AF}} \subset \mathcal{P}^{\mathrm{AS}}.$$

In fact, Example 5.3 shows that in general the first inclusion is strict. The last inclusion is always strict: On the one hand, if $b \neq 0$, then $0 \notin \mathcal{P}^{\mathrm{AF}}$, but we always have $0 \in \mathcal{P}^{\mathrm{AS}}$, on the other hand, if $b = 0$ then $\mathcal{P}^{\mathrm{AF}} = \{0\}$, but we can always set a single direction to 1 in $\mathcal{P}^{\mathrm{AS}}$. Moreover, we have the following special case.

Lemma 5.5. *If there are no potential bounds then $\mathcal{P}^{UPF} = \mathcal{P}^{AF}$ holds.*

Proof. If suffices to show $\mathcal{P}^{\mathrm{AF}} \subseteq \mathcal{P}^{\mathrm{UPF}}$. To this end, suppose first that $b = 0$. Then the only acyclic flow satisfying (5.2a) is $x = 0$. Thus, if $x = 0$ satisfies the flow bounds (5.2d), $z^+ = z^- = 0$ is the only possible solution and $\mathcal{P}^{\mathrm{UPF}} = \mathcal{P}^{\mathrm{AF}} = \{0\}$ holds.

Otherwise, let $b \neq 0$ be balanced and consider any $(z^+, z^-) \in \mathcal{P}^{\mathrm{AF}} \cap \{0,1\}^{2\mathcal{A}}$. Furthermore, let $x \in \mathbb{R}^{\mathcal{A}}$ be an acyclic flow, such that (x, z^+, z^-) satisfies the defining constraints (5.5) of $\mathcal{P}^{\mathrm{AF}}$. To show that $(z^+, z^-) \in \mathcal{P}^{\mathrm{UPF}}$ holds, we have to show that there exist potentials $\pi \in \mathbb{R}^{\mathcal{V}}$ which satisfy $\pi_u - \pi_v = \beta_a\, \psi_a(x_a)$ for all arcs $a = (u, v)$ with appropriately chosen resistances $\beta \in \overline{\mathbb{R}}^{\mathcal{A}}_{>0}$.

We first compute potentials π such that $\pi_u - \pi_v \geq x_a$ if $x_a > 0$ and $\pi_u - \pi_v \leq x_a$ if $x_a < 0$ holds for all arcs $a = (u, v) \in \mathcal{A}$. To this end, consider the digraph

$$\mathcal{D}' = \big(\mathcal{V} \cup \{r\}, \mathcal{A}(x) \cup \{(r, v) \,:\, v \in \mathcal{V}\}\big).$$

with an artificial node r and additional arcs (r, v) for all $v \in \mathcal{V}$. Furthermore, we define weights w on this graph as follows. We define $w_a = -x_a$ for arcs $a \in \mathcal{A}(x) \cap \mathcal{A}$, $w_a = x_a$ for arcs $a \in \mathcal{A}(x) \cap \overleftarrow{\mathcal{A}}$ and $w_a = 0$ for the additional arcs $a = (r, v)$. Note that by construction $\mathcal{D}'$ does not contain any cycles, because x is acyclic. Thus, the Moore-Bellman-Ford algorithm computes the shortest distances π_v from r to all nodes $v \in \mathcal{V}$ with $\pi_v \leq \pi_u + w_a$ for all $a = (u, v) \in \mathcal{A}(x)$, see, e.g., Korte and Vygen [83]. By definition of the weights w, this implies $\pi_u - \pi_v \geq x_a$ if $x_a > 0$ and $\pi_u - \pi_v \leq x_a$ if $x_a < 0$ for all arcs $a = (u, v) \in \mathcal{A}$. Moreover, note that one can shift the potentials such that $\pi \geq 0$ holds.

We now choose β-values such that equation $\pi_u - \pi_v = \beta_a\, \psi_a(x_a)$ holds for each arc $a = (u, v) \in \mathcal{A}$. For an arc a with $x_a = 0$ choosing $\beta_a = \infty$ obviously works. Otherwise, let $x_a \neq 0$. If $x_a > 0$, we can choose $\beta_a \coloneqq (\pi_u - \pi_v)/\psi_a(x_a) > 0$, since $\pi_u - \pi_v \geq x_a > 0$. Analogously, if $x_a < 0$, we can choose $\beta_a \coloneqq (\pi_u - \pi_v)/\psi_a(x_a) > 0$ since $\pi_u - \pi_v \leq x_a < 0$ and $\psi_a(x_a) < 0$. $\square$

This yields the following result.

Corollary 5.6. *If no potential bounds are present, $\mathcal{P}^{PF} \subseteq \mathcal{P}^{UPF} = \mathcal{P}^{AF} \subset \mathcal{P}^{AS}$ and the two inclusions are strict in general.*

The following results justify the choice of $\beta_a = \infty$ in $\mathcal{P}^{\mathrm{UPF}}$. Therefore, given a weakly connected digraph $\mathcal{D} = (\mathcal{V}, \mathcal{A})$ and a balanced supply and demand vector $b \in \mathbb{R}^{\mathcal{V}}$, we define the set of all potential-based flows

$$\mathcal{X}_x := \left\{x \in \mathbb{R}^{\mathcal{A}} \,:\, \exists\, \beta \in \mathbb{R}^{\mathcal{A}}_{>0},\ \pi \in \mathbb{R}^{\mathcal{V}} \text{ with } (x, \pi) \text{ satisfy (5.2a), (5.2b)}\right\}.$$

We first show that in the absence of flow and potential bounds the closure of $\mathcal{X}_x$ is given by permitting $\beta_a = \infty$. Note that $\mathcal{X}_x$ is never empty.

Proposition 5.7. *Consider a weakly connected digraph $\mathcal{D} = (\mathcal{V}, \mathcal{A})$ and a balanced supply and demand vector $b \in \mathbb{R}^{\mathcal{V}}$. The closure of potential-based flows is given by*

$$\mathrm{cl}(\mathcal{X}_x) = \left\{x \in \mathbb{R}^{\mathcal{A}} : \exists\, \beta \in \overline{\mathbb{R}}^{\mathcal{A}}_{>0},\ \pi \in \mathbb{R}^{\mathcal{V}} \textit{ with } (x, \pi) \textit{ satisfying } \text{(5.2a), (5.2b)}\right\}. \tag{5.7}$$

Proof. In the case $b = 0$, both sets only contain $x = 0$. Thus, it suffices to consider the case $b \neq 0$. In the following, we denote the set on the right-hand side of (5.7) by $\mathcal{X}^{\infty}$.

We first show $\mathrm{cl}(\mathcal{X}_x) \subseteq \mathcal{X}^{\infty}$. We assume there exists $x^* \in \mathrm{cl}(\mathcal{X}_x) \setminus \mathcal{X}_x$ as otherwise there is nothing to show. To prove $x^* \in \mathcal{X}^{\infty}$, we have to construct resistances $\beta \in \overline{\mathbb{R}}^{\mathcal{A}}_{>0}$ and potentials π^* such that (x^*, π^*) satisfy (5.2a) and (5.2b). If x^* is acyclic, we can use the procedure in the proof of Lemma 5.5. Thus, we have to show that $\mathcal{A}(x^*)$ is acyclic.

Suppose that $\mathcal{A}(x^*)$ contains a directed cycle C and w.l.o.g. assume that $x^*_a > 0$ for all $a \in C$. Let $(x^k)_{k \in \mathbb{N}} \subset \mathcal{X}_x$ converge to x^*. Then for some k_0, $x^{k_0}_a > 0$ holds for all $a \in C$, which contradicts x^{k_0} being a potential-based flow. Hence, x^* is acyclic and we conclude that the inclusion $\mathrm{cl}(\mathcal{X}_x) \subseteq \mathcal{X}^{\infty}$ holds.

We now show the reverse inclusion $\mathcal{X}^{\infty} \subseteq \mathrm{cl}(\mathcal{X}_x)$. Let $x^* \in \mathcal{X}^{\infty}$ with corresponding resistances $\beta^* \in \overline{\mathbb{R}}^{\mathcal{A}}_{>0}$ and potentials π^*. If $\beta^* \in \mathbb{R}^{\mathcal{A}}_{>0}$, then we are done due to definition. Therefore, assume that $\beta_a = \infty$ for at least one arc $a \in \mathcal{A}$. We have to construct a sequence $(x^k)_{k \in \mathbb{N}} \subset \mathcal{X}_x$ converging to x^* to finish the proof. We construct this sequence as follows:

1. We choose any sequence $(\alpha^k)_{k \in \mathbb{N}} \subset \mathbb{R}_{>0}$ with $\alpha^k \to \infty$.
2. We define a sequence of resistances $(\beta^k)_{k \in \mathbb{N}} \subset \mathbb{R}^{\mathcal{A}}_{>0}$, by using $\beta^k_a = \beta^*_a$ for all arcs a with $\beta^*_a < \infty$ and $\beta^k_a = \alpha^k$ otherwise.
3. Choose a source $s \in \mathcal{V}$ and fix $\pi^k_s = \pi^*_s$ for all $k \in \mathbb{N}$.

4. Since there are no flow or potential bounds, there exists a unique solution (x^k, π^k) of (5.2a) and (5.2b) with resistances β^k for every $k \in \mathbb{N}$. Note that for this solution x^k satisfies $x^k \in \mathcal{X}_x$.

We claim that the sequence x^k constructed this way converges to x^*. To see this, consider the subgraph $\mathcal{D}^\infty = (\mathcal{V}^\infty, \mathcal{A}^\infty)$ of $\mathcal{D}$, which results from removing all arcs with $\beta^*_a = \infty$ and all resulting isolated nodes.

Due to flow conservation and the acyclicity of potential-based flows, we can derive lower and upper bounds $\underline{x}^*$ and $\overline{x}^*$ on the flow on each arc, e.g., the flows are bounded by the total sum of inflows ($b_v > 0$). We point out that these flow bounds only depend on the graph $\mathcal{D}$ and the vector b, i.e., they are independent of the iteration k. Since the resistances β_a of arcs $a \in \mathcal{A}^\infty$ are independent of the iteration as well, the flow bounds together with (5.2b), define a lower and an upper bound on the potential difference between the nodes of each weakly connected component in $\mathcal{D}^\infty$, which have to be satisfied for all (x^k, π^k). The bounds on the potential differences can, for example, be derived as follows: For every two nodes u and v in a connected component of $\mathcal{D}^\infty$ there is – w.l.o.g. due to possible reorientation – a directed path $P \subset \mathcal{A}^\infty$ from u to v. Then $\pi_u - \pi_v$ is bounded by

$$\sum_{a \in P} \beta_a \psi_a(\underline{x}^*_a) \leq \pi_u - \pi_v \leq \sum_{a \in P} \beta_a \psi_a(\overline{x}^*_a).$$

Let $a = (u, v) \in \mathcal{A} \setminus \mathcal{A}^\infty$ and suppose that x^k_a does not converge to 0. Then (a subsequence of) the sequence $\beta^k_a\, \psi_a(x^k_a)$ converges to $\pm\infty$. If u and v are in the same connected component of $\mathcal{D}^\infty$, this contradicts the potential differences of each component being bounded. Otherwise, note that due to flow conservation there can only be flow from one connected component to another, if there is also flow to the first component from another component, and vice versa. Thus, there exists a "cycle" of arcs in $\mathcal{A} \setminus \mathcal{A}^\infty$ connecting different connected components of $\mathcal{D}^\infty$. Hence, if the flow on these arcs does not converge to 0, the potential differences of the nodes where the flow enters and leaves the different components converges to $\pm\infty$, which is a contradiction as before. Therefore, $x^k \to x^*$ holds, which concludes the proof. □

The previous result can also be extended to the case with flow bounds.

Corollary 5.8. *Using the assumptions of Proposition 5.7, the following holds. Given flow bounds* $\underline{x} \leq \overline{x}$*, we define* $\mathcal{X}_{[x]} := \mathcal{X}_x \cap [\underline{x}, \overline{x}]$*. If* $\mathcal{X}_{[x]} \neq \varnothing$*, then the closure of* $\mathcal{X}_{[x]}$ *is given by*

$$\mathrm{cl}(\mathcal{X}_{[x]}) = \{x \in \mathbb{R}^{\mathcal{A}} : \exists \beta \in \overline{\mathbb{R}}^{\mathcal{A}}_{>0},\ \pi \in \mathbb{R}^{\mathcal{V}} \textit{ with } (x, \pi) \textit{ satisfying } (5.2a),\ (5.2b),\ (5.2d)\}.$$

Proof. The inclusion "$\subseteq$" holds by the same arguments as before. To see "$\supseteq$", we use the same construction to define the sequence $(x^k, \pi^k)_{k \in \mathbb{N}}$. After possibly choosing a subsequence, all elements of the sequence either satisfy the flow bounds, or all violate the flow bounds. That is, the sequence is either contained in $\mathrm{cl}(\mathcal{X}_{[x]})$ or $\mathrm{cl}(\mathcal{X}_x) \setminus \mathrm{cl}(\mathcal{X}_{[x]})$. In the first case, we are done. Otherwise, note that the flow bounds are satisfied in the limit and thus the limit is not contained in the complement of $\mathrm{cl}(\mathcal{X}_{[x]})$ but in the intersection of the closures. □

Remark 5.9.

- Combining Lemma 5.5 and Corollary 5.8 yields that in the absence of potential bounds, acyclic flows coincide with the closure of potential-based flows.
- Instead of taking the closure of the flows only, we could also consider

  $$\mathcal{X}_{(x,\pi)} := \{(x, \pi) \in \mathbb{R}^{\mathcal{A}} \times \mathbb{R}^{\mathcal{V}} : \exists\, \beta \in \mathbb{R}^{\mathcal{A}}_{>0} \text{ s.t. } (x, \pi) \text{ satisfy (5.2a), (5.2b)}\}.$$

  Then the closure satisfies

  $$\mathrm{cl}(\mathcal{X}_{(x,\pi)}) \subsetneq \big\{(x, \pi) \in \mathbb{R}^{\mathcal{A}} \times \mathbb{R}^{\mathcal{V}} : \exists\, \beta \in \overline{\mathbb{R}}^{\mathcal{A}}_{\geq 0} \text{ s.t. } (x, \pi) \text{ satisfy (5.2a), (5.2b)}\big\},$$

  where additionally $\beta_a = 0$ is permitted. Here, the reverse inclusion is in general not true, because when using $\beta_a = 0$, flow in a cycle is possible, while potential-based flows are always acyclic.
- Note that taking the closure of potential-based flows together with the corresponding directions defined by (5.3) does *not* yield the same results as defining the directions after taking the closure of the flows, e.g., consider Figure 5.2. We have seen that $x_1 > 0$ for all $\beta_1 \in \mathbb{R}_{>0}$, and thus $z_1^+ = 1$. But by taking the closure of flows with the direction variables $x_1 = 0$ is possible, while still $z_1^+ = 1$ holds for all elements of the closure, that is, (5.3) is violated.
- For energy networks $\beta_a = \infty$ can be interpreted as if the arc is combined with a switch/valve which is turned off/closed.

5.3.2 Acyclic Subgraphs and Computational Complexity

We next obtain a complete description of $\mathcal{P}^{\mathrm{AS}}$ if the graph is planar by using known results from the literature. Indeed, acyclic $(z^+, z^-) \in \{0,1\}^{2\mathcal{A}}$ correspond to acyclic subgraphs of the digraph $\overleftrightarrow{\mathcal{D}} = (\mathcal{V}, \overleftrightarrow{\mathcal{A}})$. The corresponding *acyclic subgraph problem* was investigated by Grötschel et al. [50]. The acyclic subgraph polytope is the convex hull of incidence vectors of acyclic arc sets in a given digraph. Grötschel et al. showed that for planar graphs a complete description of the acyclic subgraph polytope is given by the variable bounds and so-called *dicycle inequalities*. These inequalities are based on the set of all dicycles (directed cycles) in $\overleftrightarrow{\mathcal{D}}$:

$$\mathcal{C} := \{C \subseteq \overleftrightarrow{\mathcal{A}} \,:\, C \text{ directed cycle}\}.$$

Note that the anti-parallel arcs $\{a, \overleftarrow{a}\}$ form a particular dicycle in $\overleftrightarrow{\mathcal{D}}$. Thus, translated to our setting, we obtain the following:

Corollary 5.10 (Grötschel et al. [50])**.** *If $\overleftrightarrow{\mathcal{D}}$ is planar then*

$$\mathcal{P}^{AS} = \Big\{(z^+, z^-) \in [0,1]^{2\mathcal{A}} \,:\, \sum_{a \in C} z_a^+ + \sum_{\overleftarrow{a} \in C} z_a^- \leq |C| - 1 \quad \forall\, C \in \mathcal{C}\Big\}. \tag{5.8}$$

Note that planarity is a reasonable assumption for real world physical networks. In general networks, however, optimizing over $\mathcal{P}^{\mathrm{AS}}$ (in fact, over all four polytopes) is NP-hard.

Lemma 5.11. *Linear optimization over $\mathcal{P}^{AS}$ is NP-hard.*

Proof. Grötschel et al. [50] already observed that linear optimization over the acyclic subgraph polytope is NP-hard, since finding a maximum acyclic subgraph is NP-hard – this problem is complementary to the feedback arc set problem, which has been proven to be NP-hard by Karp [78]. We note the graph in the reduction is simple and does not contain anti-parallel arcs. When optimizing over $\mathcal{P}^{\mathrm{AS}}$ and considering $\overleftrightarrow{\mathcal{D}}$, we can choose the weight to be 0 for either a or $\overleftarrow{a}$, depending on which direction is present in the original digraph. Thus, optimization over the acyclic subgraph polytope is equivalent to optimization over $\mathcal{P}^{\mathrm{AS}}$. ☐

Lemma 5.12. *Given a directed graph with flow bounds, and a supply and demand vector b, it is NP-complete to decide whether there exists an acyclic b-flow.*

Proof. If there exists an acyclic flow, there exists one with polynomial encoding length in the size of the instance. Moreover, acyclicity can be checked in polynomial time. Thus, the problem is in NP.

Consider an instance of the independent set problem: Given an undirected graph $G = (V, E)$ and an integer K, the question is whether there exists an independent subset of nodes of size at least K, i.e., no two selected nodes are connected by an edge. Construct the following directed graph $\mathcal{D} = (\mathcal{V}, \mathcal{A})$. The node set is

$$\mathcal{V} = \{s, t\} \cup \{v' \,:\, v \in V\} \cup \{v'' \,:\, v \in V\},$$

where s and t are two new nodes and v', v'' are distinct copies of $v \in V$. The arcs in $\mathcal{A}$ are constructed as follows: For each edge $\{u, v\} \in E$ we add two arcs (u'', v')

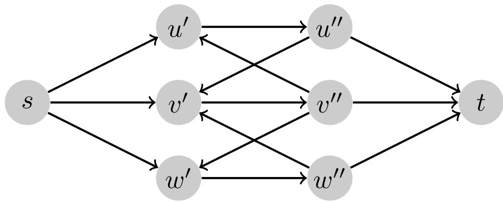


Figure 5.3. Example of the digraph $\mathcal{D} = (\mathcal{V}, \mathcal{A})$ constructed in the proof of Lemma 5.12 for the graph $G = (\{u, v, w\}, \{\{u, v\}, \{v, w\}\})$.

and (v'', u'); the corresponding flow bounds are such that the flow on these arcs is fixed to 1. See Figure 5.3 for an example of this construction. Moreover, for each node $v \in V$, we add arcs (v', v'') as well as arcs (s, v') and (v'', t); the flow on these arcs is restricted to lie in $[0, 1]$. Moreover, we set $b_{v''} = -b_{v'} = \deg(v)$ for each original node $v \in V$ and $b_s = -b_t = K$.

Note that because of the flow bounds, the direction of the flows is fixed. However, x_a can still be 0 on arcs a of type (s, v'), (v', v'') or (v'', t).

Consider an independent set $S \subseteq V$ of size K in G. Then there exists an acyclic flow in $\mathcal{D}$: For each $v \in S$, the arcs (s, v'), (v', v''), and (v'', t) have a flow value of 1. The arcs (s, v'), (v', v''), and (v'', t) for $v \notin S$ have flow 0. The flow on all other arcs is fixed to 1. It is easy to see that this forms a b-flow. Moreover, it is acyclic. Indeed, because of the flow bounds, the only directed cycles are $(v_1', v_1'', v_2', v_2'', v_1')$ for an edge $\{v_1, v_2\} \in E$ or $(v_1', v_1'', v_2', v_2'', \dots, v_j', v_j'', v_1')$ for a cycle $(v_1, v_2, \dots, v_j, v_1)$ in $G = (V, E)$. Since S is independent, the flow on either (v_1', v_1'') or (v_2', v_2'') is 0. Thus, in each cycle there is at least one arc with zero flow.

Conversely, let $x \in \mathbb{R}^{\mathcal{A}}$ be a feasible acyclic b-flow and define the following set of nodes $S \coloneqq \{v \in V : x_{(v', v'')} > 0\}$. Because of the demand of $-K$ at t and the flow bounds, we have $|S| \geq K$. Moreover, S is independent. Indeed, if there would exist an edge $\{u, v\} \subseteq S$, then x would contain a cycle (u', u'', v', v'', u'). $\square$

As a consequence, we cannot expect to obtain tractable linear descriptions for $\mathcal{P}^{\mathrm{AS}}$ and $\mathcal{P}^{\mathrm{AF}}$ in general graphs.

Obviously, acyclic subgraphs are not an accurate model for the feasible flow directions, for instance, since proper disconnected subgraphs might not even support a feasible flow. Nevertheless, the acyclic subgraph polytope is well investigated and provides a relaxation through the dicycle inequalities. We will test their practical influence on the optimization of gas networks in Section 5.4.

5.3.3 Acyclic Subgraphs with Sources and Sinks

To obtain a polytope contained in $\mathcal{P}^{\mathrm{AS}}$, but closer to $\mathcal{P}^{\mathrm{AF}}$, we consider a potential-based flow $x \in \mathcal{X}_x$ and derive valid inequalities for $\mathcal{P}^{\mathrm{AF}}$ by using the knowledge of sources and sinks in the network. For $b \in \mathbb{R}^{\mathcal{V}}$, we define the set of sources $\mathcal{V}_+ := \{v \in \mathcal{V} : b_v > 0\}$, the set of sinks $\mathcal{V}_- := \{v \in \mathcal{V} : b_v < 0\}$, and the inner nodes $\mathcal{V}_0 := \{v \in \mathcal{V} : b_v = 0\}$. Then $\mathcal{V} = \mathcal{V}_+ \mathbin{\dot\cup} \mathcal{V}_- \mathbin{\dot\cup} \mathcal{V}_0$. Moreover, for some arc set $S \subseteq \mathcal{A}$ we use the shorthand notation $z^+(S) = \sum_{a \in S} z_a^+$ and $z^-(S) = \sum_{a \in S} z_a^-$.

For every source $s \in \mathcal{V}_+$, there has to exist at least one arc with flow away from the source s. That is, there has to exists an arc $a \in \delta^+(s)$ with $x_a > 0$ or an arc $a' \in \delta^-(s)$ with $x_{a'} < 0$. Thus, $z_a^+ = 1$ or $z_{a'}^- = 1$ has to hold for at least one arc $a \in \delta^+(s)$ or $a' \in \delta^-(s)$. Similarly, for $t \in \mathcal{V}_-$, there exists at least one arc with flow towards t. This can be expressed via the valid inequalities

$$z^+\big(\delta^+(s)\big) + z^-\big(\delta^-(s)\big) \geq 1, \tag{5.9a}$$
$$z^-\big(\delta^+(t)\big) + z^+\big(\delta^-(t)\big) \geq 1. \tag{5.9b}$$

Furthermore, for every inner node $v \in \mathcal{V}_0$ in- and outflow have to be balanced, because of (5.2a). That is, if there is flow to v, i.e., there is an arc $a \in \delta^-(v)$ with $x_a > 0$ or an arc $a \in \delta^+(v)$ with $x_a < 0$, there has to exist flow from v to another node, i.e., there is an arc $a' \in \delta^-(v)$ with $x_{a'} < 0$ or an arc $a' \in \delta^+(v)$ with $x_{a'} > 0$, and vice versa. Thus, for the binary variables this implies that if there is an arc $(u, v) \in \mathcal{A}(z^+, z^-)$, there has to exist another node w with an incident arc $(v, w) \in \mathcal{A}(z^+, z^-)$, and vice versa. There are several possibilities to represent this by linear inequalities. We introduce two options, which differ in their strength and number of added inequalities.

Given a node $v \in \mathcal{V}_0$, the first option is to add an inequality for both directions of every arc incident to v. The inequalities are

$$z_a^+ \leq z^-\big(\delta^+(v) \setminus \{a\}\big) + z^+\big(\delta^-(v)\big) \qquad \forall\, v \in \mathcal{V}_0,\ a \in \delta^+(v), \tag{5.10a}$$
$$z_a^- \leq z^+\big(\delta^-(v) \setminus \{a\}\big) + z^-\big(\delta^+(v)\big) \qquad \forall\, v \in \mathcal{V}_0,\ a \in \delta^-(v), \tag{5.10b}$$
$$z_a^- \leq z^+\big(\delta^+(v) \setminus \{a\}\big) + z^-\big(\delta^-(v)\big) \qquad \forall\, v \in \mathcal{V}_0,\ a \in \delta^+(v), \tag{5.10c}$$
$$z_a^+ \leq z^-\big(\delta^-(v) \setminus \{a\}\big) + z^+\big(\delta^+(v)\big) \qquad \forall\, v \in \mathcal{V}_0,\ a \in \delta^-(v). \tag{5.10d}$$

Here, the first two inequalities imply that if arc a incident to v is oriented away from v, then there has to exist another arc that is oriented towards v. The other two inequalities imply the converse. This discrete representation of flow conservation requires $2 \sum_{v \in \mathcal{V}_0} \deg(v)$ inequalities.

Another option is to aggregate the first two inequalities and the last two inequalities, which yields

$$z^{+}\big(\delta^{+}(v)\big)+z^{-}\big(\delta^{-}(v)\big)\leq(\deg(v)-1)\Big(z^{-}\big(\delta^{+}(v)\big)+z^{+}\big(\delta^{-}(v)\big)\Big),\tag{5.11a}$$

$$z^{-}\big(\delta^{+}(v)\big)+z^{+}\big(\delta^{-}(v)\big)\leq(\deg(v)-1)\Big(z^{+}\big(\delta^{+}(v)\big)+z^{-}\big(\delta^{-}(v)\big)\Big).\tag{5.11b}$$

Again the first inequality implies that if there is an outgoing arc of node v, there has to exist an incoming arc, while the second inequality implies the converse. This representation usually has fewer inequalities: $2\,|\mathcal{V}_0|$ instead of $2\sum_{v\in\mathcal{V}_0}\deg(v)$. However, while both variants allow for the same integral points, the following example shows that they differ in the strength of their LP-relaxations.

Example 5.13. Consider again the graph shown in Figure 5.2. Here, $z_1^+=1$, $z_3^+=\frac{1}{2}$, $z_5^+=1$, and the remaining variables equal to 0 is feasible for the inequalities (5.11). In fact, it is a vertex of the LP-relaxation of (5.8) with the additional constraints (5.9) and (5.11). However, this solution is not feasible for (5.10). Furthermore, all feasible solutions of (5.10) are feasible for (5.11). This shows that the first option yields tighter LP-relaxations.

Note that for this and subsequent examples we used polymake [5, 43] to compute vertices, dimension, affine hull, and facets of polytopes.

In the following we will therefore concentrate on (5.10) and investigate the formulation given by these inequalities as well as the dicycle inequalities

$$\sum_{a\in C}z_a^{+}+\sum_{\bar{a}\in C}z_a^{-}\leq|C|-1\quad\forall\,C\in\mathcal{C}.\tag{5.12}$$

We define the *polytope of acyclic flows with sources and sinks* as

$$\mathcal{P}^{\mathrm{AS}\pm}:=\operatorname{conv}\left\{(z^{+},z^{-})\in\{0,1\}^{2\mathcal{A}}\,:\,(z^{+},z^{-})\text{ is feasible for (5.9), (5.10), (5.12)}\right\}.$$

We will also need the LP-relaxation corresponding to $\mathcal{P}^{\mathrm{AS}\pm}$:

$$\mathcal{P}^{\mathrm{AS}\pm}_{\mathrm{LP}}:=\left\{(z^{+},z^{-})\in[0,1]^{2\mathcal{A}}\,:\,(z^{+},z^{-})\text{ satisfying (5.9), (5.10), (5.12)}\right\}.$$

The model derived here turns out to be equivalent to the one investigated by Becker and Hiller [7, 8, 68], which can be seen using the analysis in the next section.

5.3.4 Analysis of Acyclic Subgraphs with Sources and Sinks

In this section, we analyze the polytope $\mathcal{P}^{\mathrm{AS}\pm}$. We start by proving a key insight, which helps to derive several results in the following. Note that we define paths to be *simple*, i.e., no node appears twice in the path. We call a directed path a *source-sink-path* if it starts in $\mathcal{V}_+$ and ends in $\mathcal{V}_-$.

Proposition 5.14. *Let $(z^+, z^-) \in \mathcal{P}^{AS\pm} \cap \{0,1\}^{2\mathcal{A}}$. For every arc in $\mathcal{A}(z^+, z^-)$, there exists a directed source-sink-path containing this arc. In particular, $\mathcal{D}(z^+, z^-)$ contains at least one path leaving each node in $\mathcal{V}_+$ and at least one path entering each node of $\mathcal{V}_-$. Moreover, if $b = 0$, then $\mathcal{P}^{AS\pm} = \{0\}$.*

Proof. Consider an arbitrary arc $a = (u, v) \in \mathcal{A}(z^+, z^-)$. To find a source-sink-path containing a, we construct a path from v to $\mathcal{V}_-$ if $v \notin \mathcal{V}_-$ and a path from $\mathcal{V}_+$ to u if $u \notin \mathcal{V}_+$. Together with the arc a, these paths yield the desired source-sink-path. Note that these two paths and their combination with a are necessarily simple, since otherwise $\mathcal{A}(z^+, z^-)$ contains a cycle.

Starting at node $v \notin \mathcal{V}_-$, by the constraints (5.9a) if $v \in \mathcal{V}_+$, or (5.10c) and (5.10d) if $v \in \mathcal{V}_0$, there exists an arc $a_1 = (v, v_1) \in \mathcal{A}(z^+, z^-)$ for some node $v_1 \in \mathcal{V} \setminus \{v\}$. Repeating this argument produces a path $(v, v_1, v_2, \ldots, v_k)$ in $\mathcal{A}(z^+, z^-)$ until it ends with $v_k \in \mathcal{V}_-$. This process terminates, since the graph is finite and we cannot produce cycles. Similarly, going backwards from $u \notin \mathcal{V}_+$, by (5.9b), (5.10a) and (5.10b) there exists an arc $(u_1, u) \in \mathcal{A}(z^+, z^-)$ for some $u_1 \in \mathcal{V} \setminus \{u\}$. Repeating yields a path $(u_r, \ldots, u_1, u)$ until $u_r \in \mathcal{V}_+$.

If $b \neq 0$, by (5.9a) there exists at least one path leaving each node in $\mathcal{V}_+$ and by (5.9b) there exists at least one path entering each node in $\mathcal{V}_-$.

Finally, let $b = 0$. Then there are no sources and sinks, i.e., the construction above would either terminate at a node with degree 1 or produce a cycle, which contradicts either (5.10) or (5.12). Thus, $\mathcal{P}^{\mathrm{AS}\pm} = \{0\}$ if $b = 0$. ☐

We obtain the following first consequence:

Corollary 5.15. *For every balanced supply and demand vector b the polytopes $\mathcal{P}^{AF}$ and $\mathcal{P}^{AS\pm}$ satisfy the inclusion*

$$\mathcal{P}^{AF} \subseteq \mathcal{P}^{AS\pm}.$$

Proof. In the case $b = 0$, we have $\mathcal{P}^{\mathrm{AS}\pm} = \mathcal{P}^{\mathrm{AF}} = \{0\}$. Hence, we only have to consider the case $b \neq 0$. Furthermore, note that it suffices to prove the inclusion for all integer points.

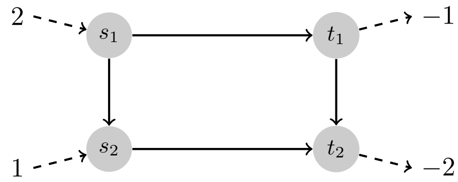


Figure 5.4. The graph shows that Theorem 5.16 does not hold in general if both $|\mathcal{V}_+| \geq 2$ and $|\mathcal{V}_-| \geq 2$: Suppose that b_{s_2}, $b_{t_2} \neq 0$ and $b_{s_2} < -b_{t_2}$, then the flow on at least one of the arcs (s_1, s_2) and (t_1, t_2) has to be positive. Nevertheless, if $b_{t_1} \neq 0$, they need not be used in $\mathcal{P}^{\mathrm{AS}\pm}$, i.e., $\mathcal{P}^{\mathrm{AS}\pm} \not\subseteq \mathcal{P}^{\mathrm{AF}}$.

Let $(z^+, z^-) \in \mathcal{P}^{\mathrm{AF}} \cap \{0,1\}^{2\mathcal{A}}$ with corresponding acyclic b-flow $x \in \mathbb{R}^{\mathcal{A}}$. By assumption $b \neq 0$, flow x is nonzero. This implies that there is at least one path with nonzero flow leaving each node in $\mathcal{V}_+$ and at least one path with nonzero flow entering each node in $\mathcal{V}_-$. Thus, the constraints (5.9) are satisfied. Due to flow conservation, the inequalities (5.10a) – (5.10d) hold. Moreover, since x is acyclic (5.12) is satisfied. Hence, we have $(z^+, z^-) \in \mathcal{P}^{\mathrm{AS}\pm}$ which concludes the proof. □

Figure 5.4 shows that $\mathcal{P}^{\mathrm{AF}} = \mathcal{P}^{\mathrm{AS}\pm}$ does not hold in general since $\mathcal{P}^{\mathrm{AS}\pm}$ does not capture the amount of supply or demand. However, the following result characterizes a special case such that the equality holds.

Theorem 5.16. *Suppose that $|\mathcal{V}_+| = 1$ or $|\mathcal{V}_-| = 1$ if $b \neq 0$. Then, if there are no flow bounds,*

$$\mathcal{P}^{AS\pm} = \mathcal{P}^{AF}.$$

Proof. In the case $b = 0$, we have $\mathcal{P}^{\mathrm{AS}\pm} = \mathcal{P}^{\mathrm{AF}} = \{0\}$. Hence, we only have to consider the case $b \neq 0$. By the previous corollary it suffices to prove that the inclusion $\mathcal{P}^{\mathrm{AS}\pm} \subseteq \mathcal{P}^{\mathrm{AF}}$ is true. Note again that it suffices to prove the inclusion for all integer points. Moreover, we only consider the single sink case $\mathcal{V}_- = \{t\}$, since the single source case is analogous.

Let $(z^+, z^-) \in \mathcal{P}^{\mathrm{AS}\pm} \cap \{0,1\}^{2\mathcal{A}}$ and denote the sources with $\mathcal{V}_+ = \{s_1, \dots, s_k\}$. We first construct an acyclic flow $x \in \mathbb{R}^{\mathcal{A}}$ with $x_a < 0$ if $z^-_a = 1$, $x_a > 0$ if $z^+_a = 1$, and $x_a = 0$ otherwise. By scaling, we will obtain a b-flow.

We start with the zero-flow $x = 0$ and define $\mathcal{P}_1 = \cdots = \mathcal{P}_k = \varnothing$. We then pick an arc $a' \in \mathcal{A}$ with either $z^+_{a'} = 1$ or $z^-_{a'} = 1$ and $x_{a'} = 0$. Then by Proposition 5.14 there exists a path P from a source s_i to the sink t in $\mathcal{D}(z^+, z^-)$ containing a' if $z^+_{a'} = 1$ or otherwise $\overleftarrow{a}'$ if $z^-_{a'} = 1$. We add P to the set $\mathcal{P}_i$ and augment x along P by one unit, by increasing x_a by 1 for every arc $a \in \mathcal{A} \cap P$ and decreasing x_a by 1 for every arc $a \in \mathcal{A}$ such that $\overleftarrow{a} \in P$. For $a \in \mathcal{A}$ we define $\Delta(P)_a = 1$ if $a \in P$ and $\Delta(P)_a = -1$ if $\overleftarrow{a} \in P$ and $\Delta(P)_a = 0$ otherwise. Then the new flow is $x + \Delta(P)$. Since $\mathcal{A}(z^+, z^-)$ does not contain both a' and $\overleftarrow{a}'$, flow $x_{a'}$ can only be increased if $a' \in \mathcal{A}(z^+, z^-)$ or otherwise decreased if $\overleftarrow{a}' \in \mathcal{A}(z^+, z^-)$ by augmenting flow along another path, even for another source-sink pair. Therefore, we can iterate augmenting flow for the remaining arcs with no flow and thereby construct a flow with the desired flow directions. Note that

$$x = \sum_{i=1}^{k} \sum_{P \in \mathcal{P}_i} \Delta(P).$$

We still have to scale the flow such that it is a b-flow. First note that there might exist an index $i \in [k] = \{1, \dots, k\}$ with $\mathcal{P}_i = \varnothing$; for example, if the graph $\mathcal{D}$ itself is a path with the ends s_1 and t. Then, if we start the procedure above with a' incident to s_1, we only augment flow once and $\mathcal{P}_i = \varnothing$ for all $i > 1$. Hence, consider $i \in [k]$ with $\mathcal{P}_i = \varnothing$. By Proposition 5.14 there exists an s_i–t-path P in $\mathcal{D}(z^+, z^-)$. Then set $\mathcal{P}_i = \{P\}$.

We now define the scaled flow

$$\sum_{i=1}^{k} \frac{b_{s_i}}{|\mathcal{P}_i|} \sum_{P \in \mathcal{P}_i} \Delta(P).$$

Since b_{s_i} and $|\mathcal{P}_i| > 0$, the scaling is valid, does not change flow directions, and yields a b-flow. □

Proposition 5.14 also helps to determine the structure of integer points in $\mathcal{P}^{\mathrm{AS}\pm}$. For a subset $S \subseteq \overleftrightarrow{\mathcal{A}}$, let $\chi(S)$ be the *incidence vector* $(\chi^+, \chi^-) \in \{0,1\}^{2\mathcal{A}}$ defined by $\chi^+_a = 1$ if $a \in S \cap \mathcal{A}$, $\chi^-_a = 1$ if $\overleftarrow{a} \in S \cap \overleftarrow{\mathcal{A}}$ and 0 otherwise.

Corollary 5.17. *Each integer point $(z^+, z^-) \in \mathcal{P}^{AS\pm}$ is the incidence vector of a union of source-sink-paths in $\overleftrightarrow{\mathcal{D}}$ that does not contain cycles and conversely.*

Proof. For each $a \in \mathcal{A}(z^+, z^-)$ there exists a source-sink-path P_a in $\mathcal{A}(z^+, z^-)$ that contains a by Proposition 5.14. Then each arc in $\mathcal{A}(z^+, z^-)$ is covered by $\cup_a P_a$ and the union does not contain cycles. Thus $(z^+, z^-) = \chi(\cup_a P_a)$.

Conversely, the incidence vectors of a union of source-sink-paths that does not contain cycles clearly satisfies (5.9), (5.10) as well as (5.12) and is therefore contained in $\mathcal{P}^{\mathrm{AS}\pm}$. □

Note that the union of source-sink-paths can contain cycles, even in the single source and sink case, see the example in Figure 5.1.

Another consequence of Proposition 5.14 is that z_a^+ and z_a^- can be fixed to 0 or 1 in some cases. We first need the following definition. Recall that we assume that $\mathcal{D}$ is weakly connected and consider two arcs distinct arcs $a_1,\ a_2 \in \mathcal{A}$ such that neither is a bridge and $\mathcal{D} - \{a_1, a_2\}$ has exactly two weakly connected components $\mathcal{D}_1$ and $\mathcal{D}_2$. Then $\{a_1, a_2\}$ is called a *cut-pair*. Assume that $\mathcal{D}_2$ contains neither source nor sink and that a_1 enters and a_2 leaves $\mathcal{D}_2$ (by reorientation). We call $\mathcal{D}_2$ *input-output subgraph*. Note that $\mathcal{D}_2$ might have no arcs, in which case a_1 and a_2 form a directed path.

Lemma 5.18. *Let $\mathcal{D}$ be the given weakly connected simple digraph with sources $\mathcal{V}_+$ and sinks $\mathcal{V}_-$. Then the following holds for every $(z^+, z^-) \in \mathcal{P}^{AS\pm}$.*

1. *If there is no source-sink-path in $\overleftrightarrow{\mathcal{D}}$ containing $a \in \mathcal{A}$ ($\overleftarrow{a} \in \overleftarrow{\mathcal{A}}$) then $z_a^+ = 0$ ($z_a^- = 0$).*
2. *Let $a \in \mathcal{A}$ be a bridge, i.e., $\mathcal{D} - a$ is not weakly connected, and let the two connected components of $\mathcal{D} - a$ be induced by B_1, B_2 with $\mathcal{V} = B_1 \,\dot{\cup}\, B_2$. Then the following holds for each of the two connected components $\mathcal{D}[B_i]$.*
   a) *Assume $B_i \cap \mathcal{V}_+ = \varnothing$ and $B_i \cap \mathcal{V}_- = \varnothing$. Then $z_a^- = z_a^+ = 0$. Furthermore, $z_{a'}^- = z_{a'}^+ = 0$ holds for all arcs in the induced subgraph $a' \in \mathcal{D}[B_i]$.*
   b) *Assume $B_i \cap \mathcal{V}_+ \neq \varnothing$ and $B_i \cap \mathcal{V}_- = \varnothing$. If $a \in \delta^+(B_i)$, then $z_a^+ = 1$ and if $a \in \delta^-(B_i)$, then $z_a^- = 1$ holds.*
   c) *Assume $B_i \cap \mathcal{V}_+ = \varnothing$ and $B_i \cap \mathcal{V}_- \neq \varnothing$. If $a \in \delta^+(B_i)$, then $z_a^- = 1$ and if $a \in \delta^-(B_i)$, then $z_a^+ = 1$ holds.*
3. *Let there exist an input-output subgraph of $\mathcal{D}$ with entering arc a and leaving arc a'. Then $z_a^+ = z_{a'}^+$ and $z_a^- = z_{a'}^-$.*

Proof. In all cases, it suffices to consider integer points $(z^+, z^-) \in \mathcal{P}^{\mathrm{AS}\pm}$, since the statement then holds for the convex hull $\mathcal{P}^{\mathrm{AS}\pm}$.

1. Suppose that there is no source-sink-path in $\overleftrightarrow{\mathcal{D}}$ containing a ($\overleftarrow{a}$). By Proposition 5.14, $\mathcal{A}(z^+, z^-)$ cannot contain a ($\overleftarrow{a}$). Thus, $z_a^+ = 0$ ($z_a^- = 0$).
2. In case (2a), B_i contains neither a source nor a sink. Since the paths are simple and $\mathcal{D}(z^+, z^-)$ is acyclic, no path can enter B_i. In case (2b), at least one source-sink-path has to leave B_i. In case (2c), at least one source-sink-path has to enter B_i.
3. Every union of source-sink-paths using a_1 also has to use a_2. This implies the given equations. □

Remark 5.19.

1. The results in Lemma 5.18 are similar to the ones by Becker and Hiller [7, 8], but in a different notation.

2. The existence of a node of degree 2 is a special case of Part 3 of Lemma 5.18.
3. It is an open question whether the conditions of Lemma 5.18 define the affine hull of $\mathcal{P}^{\mathrm{AS}\pm}$ in general; we will present a result for the special single source and single sink case in Proposition 5.28.

A natural question is how the conditions in Lemma 5.18 can be checked. It turns out that the condition of Part 1 is hard to check, even for the single source and sink case.

Proposition 5.20. *Given a directed graph with source node s, sink node t and some arc $a = (u, v)$, it is NP-complete to decide whether there exists a (simple) s–t-path that contains a.*

Proof. Consider the k-vertex disjoint paths problem, which consists of finding vertex disjoint paths from s_i to t_i for a given set of node pairs $(s_1, t_1), \dots, (s_k, t_k)$. Obviously, finding an s–t-path that uses a is the special case of finding 2-vertex disjoint paths between (s, u) and (v, t). Fortune et al. [39] proved that the vertex disjoint paths problem on general directed graphs is NP-complete even for fixed $k \geq 2$. □

Corollary 5.21. *Linear optimization over $\mathcal{P}^{AS\pm}$ is NP-hard, even if there is a single source s and sink t.*

Proof. Consider the linear function that maximizes z_a^+ for some arc a over $\mathcal{P}^{\mathrm{AS}\pm}$. The optimal value is 1 if and only if there exists an s–t-path through a. The results then follows by NP-hardness of determining the latter by Proposition 5.20. □

Remark 5.22. Schrijver [130] showed that for fixed k and planar graphs, an s–t-path that contains a given arc can be found in polynomial time. Recently, Fakcharoenphol et al. [32] showed that one can compute the set of all arcs that are not contained in an s–t-path of a planar graph in linear time.

Remark 5.23. With respect to Parts 2 and 3 of Lemma 5.18 the following holds. Bridges in graphs can be found in linear time, see Tarjan [148]. Moreover, checking whether a source and sink are in the same connected component can be done by breadth-first search in linear time, see, e.g., Korte and Vygen [83]. Moreover, after bridges have been removed, the linear time algorithm of Mehlhorn et al. [100] outputs a cut-pair if one exists. More input-output subgraphs can be produced using this algorithm iteratively.

5.3.5 Analysis of the Single Source and Sink Case

In this section, we provide a further analysis for the special case of a single source s and sink t. This implies that the balanced $b \in \mathbb{R}^{\mathcal{V}}$ satisfies $b_s = -b_t \geq 0$ and $b_v = 0$ for all $v \in \mathcal{V} \setminus \{s,t\}$. To simplify notation, we orient the arcs incident to the source s and sink t such that $\delta^-(s) = \varnothing$ and $\delta^+(t) = \varnothing$ holds.

Lemma 5.24. *Let $\mathcal{D}$ be the given weakly connected simple digraph with source s and sink t. Then for every $(z^+, z^-) \in \mathcal{P}^{AS\pm}$, $z_a^- = 0$ holds for every $a \in \delta^+(s)$ and for every $a \in \delta^-(t)$.*

Proof. Since $\mathcal{P}^{\mathrm{AS}\pm}$ is the convex hull of the integer points, it suffices to prove that the statement holds for all integer points. Thus, consider an integer point $(z^+, z^-) \in \mathcal{P}^{\mathrm{AS}\pm}$. Assume the statement does not hold, and let $a = (s, v)$ with $z_a^- = 1$. Then going backwards from v similarly to the proof of Proposition 5.14 shows that there exists an s–v-path. This would close a cycle, hence, $z_a^- = 0$ holds for all $a \in \delta^+(s)$. For $a \in \delta^-(t)$ we can argue analogously. □

Remark 5.25. Let $S \subset \mathcal{V}$ with $s \in S$, $t \notin S$ and consider the *s–t-cut inequalities*

$$z^+(\delta^+(S)) + z^-(\delta^-(S)) \geq 1.$$

Because of Corollary 5.17 these inequalities are valid for all integer points in $\mathcal{P}^{\mathrm{AS}\pm}$ and thus for their convex hull. However, they are weaker than the inequalities (5.9) and (5.10). Indeed, the s–t-cut inequalities together with nonnegativity provide a complete linear description of the dominant of the s–t-path polytope, see, e.g., Schrijver [131, Theorem 13.1]. Moreover, as an example consider the incidence vector of the union of (at least) one s–t-path and some node-disjoint arc. This vector is feasible for the s–t-cut inequalities, but not for (5.9) and (5.10). However, the s–t-cut inequalities can be strengthened as follows.

Lemma 5.26. *Let $\mathcal{D}$ be a simple connected digraph with source s and sink t. Then for all subsets $S \subset \mathcal{V}$ with $s \in S$ and $t \notin S$ the inequalities*

$$z^+(\delta^+(S)) + z^-(\delta^-(S)) \geq 1 + z_a^- \qquad \forall a \in \delta^+(S) \tag{5.13a}$$

$$z^+(\delta^+(S)) + z^-(\delta^-(S)) \geq 1 + z_a^+ \qquad \forall a \in \delta^-(S) \tag{5.13b}$$

are valid for $\mathcal{P}^{AS\pm}$.

Proof. Inequality (5.13a) is satisfied by all solutions in $\mathcal{P}^{\mathrm{AS}\pm}$ with $z_a^- = 0$ by Remark 5.25. Let $(z^+, z^-) \in \mathcal{P}^{\mathrm{AS}\pm}$ be an integral solution with $z_a^- = 1$. By Proposition 5.14, $\mathcal{D}(z^+, z^-)$ contains an s–t-path P that uses $\overleftarrow{a}$. Then P has to cross the cut at least twice and therefore $z^+(\delta^+(S)) + z^-(\delta^-(S)) \geq 2$. Thus, (5.13a) holds for the convex hull of these integer points, i.e., $\mathcal{P}^{\text{AS}\pm}$. The validity of (5.13b) can be seen similarly. $\square$

Remark 5.27. Note that (5.13a) for $S = \{s\}$ yields (5.9a), since $z_a^- = 0$ for all arcs $a \in \delta^+(s)$ by Lemma 5.24.

Since we want to use $\mathcal{P}^{\text{AS}\pm}$ to speed-up the optimization of energy networks, we are interested in the affine hull of $\mathcal{P}^{\text{AS}\pm}$ to identify possible equations or variable fixings, which we can use to strengthen our problem. In Lemma 5.18 we already discussed such conditions for the general case. For the single sink and single sink case we can prove the following result concerning the affine hull of $\mathcal{P}^{\text{AS}\pm}$.

Proposition 5.28. *Assume that in $\overleftrightarrow{\mathcal{D}}$ there exist two arc-disjoint s–t-paths and for every arc $a \in \overleftrightarrow{\mathcal{A}} \setminus \{\overleftarrow{a} : a \in \delta^+(s) \cup \delta^-(t)\}$, there exists two s–t-paths that are arc-disjoint except for a. Then*

$$\dim \mathcal{P}^{AS\pm} = |\overleftrightarrow{\mathcal{A}}| - \deg(s) - \deg(t),$$

where δ^+, δ^- and deg *are with respect to the original digraph $\mathcal{D}$.*

Proof. Consider an equation $(c^+)^\top z^+ + (c^-)^\top z^- = \gamma$ that is valid for $\mathcal{P}^{\text{AS}\pm}$, i.e., we have $\mathcal{P}^{\text{AS}\pm} \subset \{(z^+, z^-) : (c^+)^\top z^+ + (c^-)^\top z^- = \gamma\}$. Let $c = (c^+, c^-)$.

By assumption there exist two arc-disjoint s–t-paths P_1 and P_2. Then the incidence vectors $\chi(P_1)$, $\chi(P_2)$, and $\chi(P_1 \cup P_2)$ are contained in $\mathcal{P}^{\text{AS}\pm}$. Thus, the equalities $c^\top\chi(P_1) = c^\top\chi(P_2) = \gamma$ and $c^\top\chi(P_1 \cup P_2) = c^\top\chi(P_1) + c^\top\chi(P_2) = \gamma$ hold. Adding the first two equations yields $c^\top\chi(P_1) + c^\top\chi(P_2) = 2\,\gamma$ which implies $\gamma = 0$.

Consider an arbitrary arc $a \in \overleftrightarrow{\mathcal{A}} \setminus \{\overleftarrow{a} : a \in \delta^+(s) \cup \delta^-(t)\}$. By assumption there exist two s–t-paths P_1 and P_2 that are arc-disjoint, except for a. Let $\hat{P}_1 \coloneqq P_1 \setminus \{a\}$ and $\hat{P}_2 \coloneqq P_2 \setminus \{a\}$. Then for $i \in \{1, 2\}$ we get

$$c^\top\chi(P_i) = c^\top\chi(\hat{P}_i) + c_a = 0 \tag{5.14}$$

and moreover

$$c^\top\chi(P_1 \cup P_2) = c^\top\chi(\hat{P}_1) + c^\top\chi(\hat{P}_2) + c_a = 0$$

holds. Adding equation (5.14) for $i \in \{1, 2\}$ yields

$$c^\top\chi(\hat{P}_1) + c^\top\chi(\hat{P}_2) + 2\,c_a = 0.$$

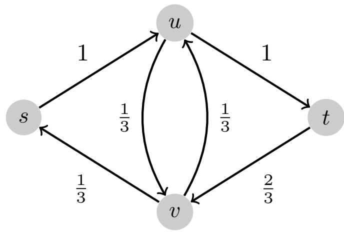


Figure 5.5. The graph $\mathcal{D}(z^+, z^-)$ associated with a vertex of $\mathcal{P}_{\mathrm{LP}}^{\mathrm{AS}\pm}$ applied to the network of Figure 5.2.

This implies $c_a = 0$, and therefore $c_a = 0$ for all arcs $a \in \overleftrightarrow{\mathcal{A}} \setminus \{\overleftarrow{a} \,:\, a \in \delta^+(s) \cup \delta^-(t)\}$. Thus, the only possible equations are the ones of Lemma 5.24, i.e., $z_a^- = 0$ for all arcs $a \in \delta^+(s) \cup \delta^-(t)$, and linear combinations of them. Since the former are linearly independent, it suffices to consider them. Together, they reduce the dimension by $|\delta^-(s)| + |\delta^+(t)|$, which shows the claim. $\square$

Note that the assumptions of Proposition 5.28 rule out bridges as well as input-output subgraphs. If $\mathcal{D}$ would contain a bridge such that s and t are in different connected components, then there are no two arc-disjoint s–t-paths. Otherwise if s and t are in the same connected component, then there is no s–t-path using the bridge. If a cut-pair would exist, then there are no two arc-disjoint paths using the arcs of the cut-pair. Moreover, note that the assumptions also rule out the existence of an s–t-arc, since the only s–t-path using this arc would be the arc itself. Nevertheless, we could allow the existence of an s–t-arc, but then, if $(s,t) \in \overleftrightarrow{\mathcal{A}}$, the formula would have to be $\dim \mathcal{P}^{\mathrm{AS}\pm} = |\overleftrightarrow{\mathcal{A}}| - \deg(s) - \deg(t) + 1$ due to double counting.

The following examples show that the defining inequalities of $\mathcal{P}^{\mathrm{AS}\pm}$ do not cover all facet defining inequalities of $\mathcal{P}^{\mathrm{AS}\pm}$. However, they all do define facets of $\mathcal{P}^{\mathrm{AS}\pm}$ in particular examples. Furthermore, after the following examples we will assess the influence of $\mathcal{P}^{\mathrm{AS}\pm}$ on the optimization of stationary gas networks in the next section.

Example 5.29. Consider again the graph given in Figure 5.2. This example shows that the LP-relaxation $\mathcal{P}_{\mathrm{LP}}^{\mathrm{AS}\pm}$ is not integral, i.e., $\mathcal{P}^{\mathrm{AS}\pm} \subsetneq \mathcal{P}_{\mathrm{LP}}^{\mathrm{AS}\pm}$. Indeed, Figure 5.5 depicts a graph associated with a vertex of $\mathcal{P}_{\mathrm{LP}}^{\mathrm{AS}\pm}$.

Example 5.30. Again consider the graph in Figure 5.2. Lemma 5.24 implies that

$$z_1^- = z_2^- = z_4^- = z_5^- = 0$$

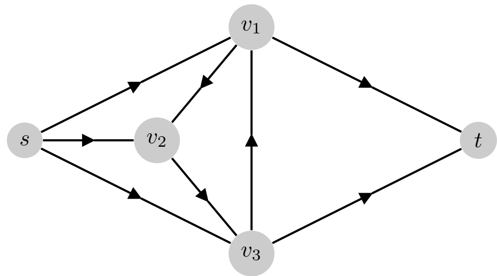


Figure 5.6. A graph for which dicycle inequalities (5.12) define facets of $\mathcal{P}^{\mathrm{AS}\pm}$.

holds for all points in $\mathcal{P}^{\mathrm{AS}\pm}$. Furthermore, the other variables can take either value and these equations define the affine hull. Thus, in this example, the dimension of $\mathcal{P}^{\mathrm{AS}\pm}$ satisfies $\dim \mathcal{P}^{\mathrm{AS}\pm} = |\overleftrightarrow{\mathcal{A}}| - |\delta^+(s)| - |\delta^-(t)|$, although the assumptions of Proposition 5.28 are not satisfied for arc 3.

Most facets of $\mathcal{P}^{\mathrm{AS}\pm}$ are already part of the defining system of $\mathcal{P}^{\mathrm{AS}\pm}_{\mathrm{LP}}$: the variable bounds $z_1^+, z_2^+, z_4^+, z_5^+ \leq 1$ and $z_3^-, z_3^+ \geq 0$, the inequalities $1 \leq z_1^+ + z_2^+$, $1 \leq z_4^+ + z_5^+$ (see (5.9a) and (5.9b)) and $z_3^- + z_3^+ \leq 1$ (see (5.12)), and the node conditions (5.10a) – (5.10d) at nodes u and v:

$$
\begin{aligned}
z_1^+ &\leq z_3^+ + z_4^+, & \qquad z_2^+ &\leq z_3^- + z_5^+,\\
z_3^- &\leq z_4^+, & z_3^- &\leq z_2^+,\\
z_3^+ &\leq z_1^+, & z_3^+ &\leq z_5^+,\\
z_4^+ &\leq z_1^+ + z_{3,}^-, & z_5^+ &\leq z_2^+ + z_3^-.
\end{aligned}
$$

Note that these inequalities differ a bit from (5.10a) – (5.10d), since we omit the variables already fixed to 0. The only facets that are given by other inequalities are

$$
\begin{aligned}
1 + z_3^- &\leq z_2^+ + z_3^+ + z_4^+,\\
1 + z_3^+ &\leq z_1^+ + z_3^- + z_5^+,
\end{aligned}
\tag{5.15}
$$

which arise from (5.13a) and (5.13b).

Example 5.31. Interestingly, in the previous example there are no facets defined by the dicycle inequalities (5.12). Note that this does not hold in general, e.g., consider the graph in Figure 5.6. Here, the dicycle inequalities defined by the two cycles $\{(v_1, v_2), (v_2, v_3), (v_3, v_1)\}$ and $\{(v_1, v_3), (v_3, v_2), (v_2, v_1)\}$ define facets of $\mathcal{P}^{\mathrm{AS}\pm}$.

5.4 Numerical Results

To demonstrate the effect of our method to handle acyclic flows via $\mathcal{P}^{\mathrm{AS}\pm}$, we describe computational experiments for stationary gas networks with the gas flow given as a potential-based flow. In the next chapter, we will also present a detailed study of the effect of using $\mathcal{P}^{\mathrm{AS}\pm}$ together with the ODE constrained model and Algorithm 4.2, which we introduced in the previous chapter. As mentioned earlier, gas networks usually contain additional active elements. For these we use the models presented in Section 4.2 in the previous chapter. In particular, for the compressor stations we use the constraints (4.16) without the additional facets. Instead of the ODE model introduced in the previous chapter, pipes are handled in the way described in Example 5.1 by

$$\pi_u - \pi_v = \beta_a \psi(x_a)$$

with $\psi(x_a) = |x_a|\, x_a$. Thereby, the resistances β_a are given by

$$\beta_a = \left(\frac{4}{\pi}\right)^2 \frac{L_a}{D_a^5}\, \lambda_a\, c^2,$$

where L_a is the length of the pipe, D_a its diameter, λ_a the friction coefficient given by (1.3), and c the speed of sound; see also Section 1.1 for a derivation of the Weymouth equation. Recall from Example 5.1 that in stationary gas transport, the potentials are the squares of the pressures at the nodes. To integrate the potential equation with the other models, we couple the potentials with the pressure variables through equations $p_u^2 = \pi_u$. Furthermore, we use variables x for the potential-based flow in this section, which correspond to the mass flow variables q in stationary gas transport.

Since some of the models of network elements other than pipes cannot be (linearly) expressed in terms of the potentials, our model includes pressure variables p_u for all nodes $u \in \mathcal{V}$. To avoid unnecessary nonlinear equations, our model only contains the potential variables π_u and the equations $p_u^2 = \pi_u$ for nodes $u \in \mathcal{V}$ where they are actually needed. Note that this is a modeling choice and one could also only introduce the pressure variables as needed. More details on the implementation can be found in Section 6.1.

Before presenting our numerical results, we briefly describe how the flow direction variables z^+, z^- are integrated into the models and how the cycles of the network are detected.

5.4.1 Model Integration of Flow Direction Variables

For handling acyclic flows, we include flow direction variables z^+, z^- in our model for all network elements except for control valves and compressor stations. Since these are one-directional elements, i.e., the flow through control valves and compressor stations is nonnegative, and their models contain binary variables z_{cv} and z_{cs} which are coupled with the flow anyway, we use these variables instead of additional flow direction variables z^+_{cv}, z^-_{cv}, z^+_{cs}, and z^-_{cs}. Of course, control valves and compressor stations are included in the binary flow direction constraints (5.9) and (5.10). Therefore, we use z_{cv} and z_{cs} as if we had introduced flow direction variables z^+_{cv}, z^-_{cv}, z^+_{cs}, and z^-_{cs} with $z^+_{cv} = z_{cv}$, $z^-_{cv} = 0$ and $z^+_{cs} = z_{cs}$, $z^-_{cs} = 0$. The treatment of control valves and compressor stations, in particular those with bypasses, in dicycle inequalities (5.12) will be discussed in Remark 5.33.

For all other network elements $a \in \mathcal{A} \setminus (\mathcal{A}^{cs} \cup \mathcal{A}^{cv})$, we add the variables z^+_a and z^-_a to their respective models. We couple z^+_a and z^-_a to the flow variables by using the linear relaxation (5.6) of (5.3), that is, we use $\underline{x}_a\, z^-_a \leq x_a \leq \overline{x}_a\, z^+_a$ instead of $\operatorname{sgn}(x_a) = z^+_a - z^-_a$. Moreover, they can be coupled to some element specific models as follows:

- Since the pressure decreases in the direction of flow over pipelines or nonlinear resistors, we can strengthen our model by including the inequality

$$(\underline{p}_u - \overline{p}_v)\, z^-_a \leq p_u - p_v \leq (\overline{p}_u - \underline{p}_v)\, z^+_a.$$

  Note that this is especially an improvement of the ODE relaxation as discussed in Section 4.6.

- If a valve is closed, i.e., $z_{va} = 0$, there is no flow. Thus, we couple the flow directions with the binary variable of the valve by

$$z^+_a + z^-_a = z_{va}.$$

- The model for linear resistors already contains indicator variables $z^{\varepsilon+}_a$, $z^{\varepsilon-}_a$ if the flow is greater/less or equal than $\pm q_\varepsilon$. These indicator variables are combined with the flow directions via the inequalities

$$z^{\varepsilon+}_a \leq z^+_a, \qquad z^{\varepsilon-}_a \leq z^-_a.$$

Remark 5.32. Note that for nodes with degree 1, we do not add the binary flow conservation constraints (5.9) and (5.10). Instead we directly fix the flow direction variables of the incident arc depending on the node being a source, sink or inner node.

The dicycle inequalities (5.12) can be added for a cycle basis or for all cycles in the network. To find the cycles we consider the underlying undirected graph, where we treat each control valve or compressor station with their respective bypass as a single edges, since we already add the inequalities (4.14) and (4.17) which forbid that both control valve/compressor station and the bypass can be active/open. Details on the treatment of control valves and compressor stations will be discussed in Remark 5.33. A cycle basis of the undirected graph is computed as follows. After computing a spanning tree by *breadth-first-search*, each non-tree edge defines a cycle which is added to the cycle basis. The possible combinations of these cycles are then enumerated. Two cycles can be combined into a new cycle, if their symmetric difference induces a connected subgraph, where all nodes have degree two. We remark that all cycles of the graph can be found by enumerating all possible combinations of basis cycles. Moreover, note that the enumeration and checking if two cycles can be combined takes only a fraction of a second for the networks considered here. Each of these cycles of the underlying undirected graph then defines two dicycles for which we add the inequalities (5.12).

Remark 5.33. In the very beginning of this chapter, we argued that potential-based flow in cycles (over pipelines) is not possible. However, we also forbid flow in cycles containing control valves or compressor stations. Since control valves reduce the the pressure in the direction of the flow, we can conclude by the same arguments as before that (gas) flow over cycles including control valves is not possible. In contrast, compressor stations increase the pressure level, which actually makes flow in cycles possible. But note that such cycles can significantly increase the temperature of the gas. This could only be controlled in (transient) models where the temperature of the gas is kept under control as well. Therefore, we decided to also forbid flow in cycles containing compressor stations.

Since control valves and compressor stations are one-directional elements, we have to take special care of these. Therefore, we distinguish two cases: control valves/compressor stations without bypass and control valves/compressor stations with bypass. In the first case, if an edge of an undirected cycle corresponds to an one-directional element, then this cycle only defines one dicycle. If two one-directional elements are part of the same cycle, but in reverse direction, then this cycle does not define a dicycle for inequalities (5.12).

In the second case, a cycle with a bypass defines two dicycles, i.e., one in both directions, and the control valve/compressor station defines another dicycle. However, since not both the bypass can be open and the control valve/compressor station can be active at the same time, we can combine the dicycle inequalities as in the following example. Consider a cycle consisting of pipe 1, a control valve *cv* with bypass *va* and pipe 2 closing the cycle. Then we add the following dicycle inequalities to our model:

$$z_1^+ + z_2^+ + z_{va}^+ + z_{cv} \leq 2,$$
$$z_1^- + z_2^- + z_{va}^- \leq 2.$$

5.4.2 Results

In order to test the effects of the different conditions we performed computations with the following model variants:

NFD	*no* binary variables to represent flow directions;
FDO	with binary variables, but no flow conservation or dicycle inequalities;
CB	dicycle inequalities (5.12) for a cycle basis;
AC	dicycle inequalities (5.12) for all cycles;
FLC	binary flow conservation (5.9) and (5.10);
FLC+CB	variant FLC plus dicycle inequalities for a cycle basis;
FLC+AC	variant FLC plus dicycle inequalities for all cycles.

We use the gas network instances `GasLib-40`, which we already used in Section 4.6, and `GasLib-582` from the library `GasLib` [41, 126]. Recall that the `GasLib-40` network has 40 nodes, 39 pipes, and 6 compressor stations. The `GasLib-582` network has 582 nodes, 278 pipes, 5 compressor stations, 23 control valves, 8 resistors, 26 valves, and 269 short pipes. In total there are 4227 different scenarios for network `GasLib-582`, arising from different distributions of the loads.

The computations were performed on a cluster with 3.5 GHz Intel Xeon E5-1620 Quad-Core CPUs, having 32 GB main memory and 10 MB cache running Linux. We used SCIP version `7.0.0` [40, 132] with a time limit of one hour and we used CPLEX version `12.10.0` as LP-solver.

As objective function for the following tests we chose the maximization of the sum of pressures. Several other possibilities for the objective function exist, see Section 6.3.2.

Since the `GasLib-40` instance is rather small, the solving time for all models is less than a second (and hence not reported in detail). Thus, this does not allow to draw conclusions on the different model variants. Nevertheless, the instance gives insight on some advantages of using the flow direction variables. Table 5.1 shows statistics for the flow bounds of the pipes after presolving. Column "#fixed flows" shows the number of pipes with fixed flow, column "#fixed dirs" the number of pipes with fixed flow direction, and columns "$\underline{x}_{mean}$" and "$\overline{x}_{mean}$" the arithmetic mean lower and upper bounds of the flows. While the number of fixed flows is the same for all models, and mainly depends on the graph and the position of sources and sinks, the

Table 5.1. Statistics for the flow bounds after presolving of 39 pipes in the network `GasLib-40` for all model variants.

variant	#fixed flows	#fixed dirs	$\underline{x}_{mean}$	$\overline{x}_{mean}$
NFD	13	2	−189.23	245.10
FDO	13	12	−100.39	115.06
CB	13	17	−61.03	74.62
AC	13	17	−60.72	74.58
FLC	13	12	−98.25	111.73
FLC+CB	13	20	−41.39	54.33
FLC+AC	13	20	−41.24	54.33

number of fixed directions and the arithmetic mean flow bounds can be improved by using the flow direction variables. This is also illustrated in Figure 5.7, which compares models NFD and FLC+AC. The figure distinguishes pipes with fixed flow, fixed direction and the remaining pipes.

Note that we do not use optimality-based bound tightening (OBBT) in our experiments with the potential-based flow model, see, e.g., Gleixner et al. [47] and the references therein. Indeed, Becker and Hiller [8] use OBBT to further strengthen flow bounds. However, we use the customized OBBT method described in Section 6.2 for the experiments with the ODE model. Note that a more detailed study of OBBT and the effect of $\mathcal{P}^{\mathrm{AS}\pm}$ on the ODE model is part of the next chapter.

The above results show that the arithmetic mean flow interval for model NFD is more than four times as large as the flow interval of model FLC+AC. Since tighter variable bounds typically lead to smaller branch-and-bound trees, this positive effect

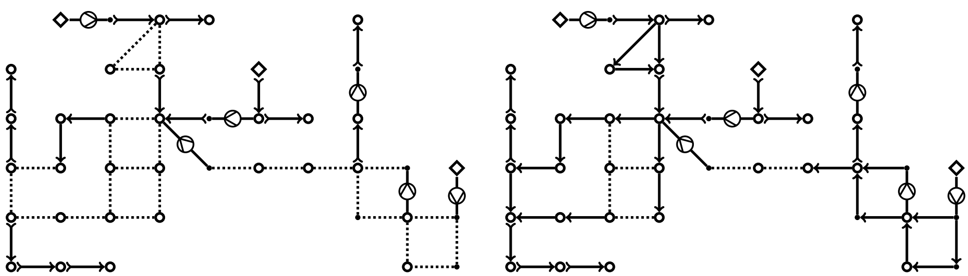


Figure 5.7. The presolved network `GasLib-40` corresponding to variants NFD (left) and FLC+AC (right). The scenario has 3 sources (diamonds) and 29 sinks (circles). Pipes with fixed flow are depicted by $\rightarrowtail$, fixed flow directions are shown by $\rightarrow$, and the remaining pipes (with unfixed flows/directions) are dashed.

Table 5.2. Aggregated results for `GasLib-582` scenarios for all model variants with the potential-based flow model.

variant	opt	feas	limit	inf	inf-presol
NFD	218	966	864	2179	2168
FDO	819	1223	6	2179	2168
CB	586	1454	8	2179	2168
AC	767	1278	3	2179	2174
FLC	1818	220	10	2179	2168
FLC+CB	1883	164	1	2179	2170
FLC+AC	2015	30	3	2179	2174

Table 5.3. Geometric means of solving times in seconds and total run time in hours for all model variants for the `GasLib-582` scenarios.

variant	to opt	to first	to inf	total	total time [h]
NFD	1435.02	892.10	1.01	50.59	1948.68
FDO	1102.95	89.46	1.01	42.32	1553.21
CB	1209.24	81.65	1.01	45.76	1713.48
AC	1313.93	81.12	1.00	44.12	1627.08
FLC	549.28	113.09	1.01	23.69	641.43
FLC+CB	465.97	103.53	1.01	21.36	537.70
FLC+AC	290.98	68.04	1.00	15.94	270.28

on the flow bounds is also reflected in the solving times of larger instances as can be seen by the following results for the `GasLib-582` network.

The results of each model variant on the `GasLib-582` network are given in Tables 5.2, 5.3, and 5.4 aggregated over all 4227 scenarios. In Table 5.2, column “opt” states the number of feasible scenarios solved to optimality, “feas” the number of scenarios for which a feasible solution could be found, but which could not be solved to optimality, “limit” the number of scenarios running into the time limit without a feasible solution, “inf” the total number of scenarios that have been determined to be infeasible, and “inf-presol” the number of infeasible scenarios for which infeasibility has been detected during presolving.

Table 5.3 provides statistics for geometric mean times in seconds and the total running time in hours of the model variants. Here, column “to opt” shows the geometric mean time it took to prove optimality, the column “to first” gives the geometric mean time until the first feasible solution was found, “to inf” the geometric mean time to prove infeasibility, “total” the total geometric mean time over all scenarios, and “total time” shows the total computational time. Thereby, note that the total

Table 5.4. Statistics on the flow bounds after presolving of 278 pipes in network `GasLib-582` for all model variants.

variant	#fixed flows	#fixed dirs	#min dirs	#max dirs	$\underline{x}_{mean}$	$\overline{x}_{mean}$
NFD	152.95	44.44	23	64	−111.03	174.98
FDO	152.41	50.35	25	83	−102.40	168.62
CB	151.87	52.81	25	87	−103.78	168.06
AC	152.43	53.14	25	104	−98.09	150.40
FLC	151.74	50.39	21	87	−103.41	166.69
FLC+CB	152.40	55.10	21	93	−94.53	162.76
FLC+AC	152.23	56.59	21	102	−84.63	135.68

geometric mean time over all scenarios is much less than the time to optimality due to the running times of the infeasible scenarios.

In Table 5.4 statistics similar to Table 5.1 are given. Here, the results are averaged over all scenarios. Additionally, the columns "#min dirs" and "#max dirs" show the minimal and maximal number of fixed flow directions over all scenarios.

Remark 5.34. The numerical results shown in Table 5.2 are consistent in the sense that all variants identify the same 2179 infeasible scenarios. Moreover, for all other scenarios feasible solutions were found by at least one model variant. In fact, only 20 scenarios could not be solved to optimality. For these scenarios, at least one model variant found a feasible solution with an optimality gap of 0.4 % or better.

The results clearly show that determining infeasibility seems to be easy in most cases. With all model variants, almost all infeasible scenarios could already be identified during presolving. That is, our acyclic flow models only slightly improve the computations here. However, the numbers of feasible scenarios show a completely different picture. Comparing the solving times in Table 5.3 for the basic model without any binary direction variables (i.e., NFD) with the model enhanced by $\mathcal{P}^{\mathrm{AS}\pm}$ (i.e., FLC+AC) shows a speed-up factor of ~ 4.9 for the geometric mean time to prove optimality, and a speed-up factor of ~ 7.2 for the total running times. Moreover, Table 5.2 shows that with model NFD no feasible solution has been found for almost half of the feasible scenarios, whereas with model FLC+AC almost all feasible scenarios could be solved to optimality.

A partial explanation for the performance improvement is as follows. Using the flow direction variables and the additional constraints, we can represent properties of feasible solutions, which are otherwise not included in the initial model or relaxation. Moreover, the heuristics and presolving techniques can detect more variable fixings, implications and reductions based on the flow conservation and the flow direction variables. For example, in diving heuristics, it is easier to detect infeasibility based on the binary flow direction variables without having to consider the nonlinear physics. This leads to tighter variable bounds (after presolving), which can be seen for the flow variables in Tables 5.1 and 5.4. Since having tight variables bounds is important to derive good relaxations for the nonlinearities, the LP-relaxations are already stronger early in the branch-and-bound tree. Moreover, tighter variable bounds (typically) lead to smaller branch-and-bound trees, since the search space is smaller. Indeed, we can observe this effect for the computations on the network `GasLib-582`. The arithmetic mean number of branch-and-bound nodes (rounded up) for the feasible scenarios solved to optimality with model variant NFD is 168 970, while it is 35 330 with model FLC+AC. That is, in model NFD it takes about 4.8-times as many nodes in comparison with variant FLC+AC, which is almost the same ratio as the speed-up for the geometric mean solving time to optimality shown in Table 5.3.

Another interesting observation is that variants AC and FLC form an exception of the positive effect of having tighter relaxations. Tables 5.1 and 5.4 both suggest that model variant AC defines the tighter LP-relaxation. Furthermore, also the geometric mean time until the first solution is found for the `GasLib-582` scenarios is smaller with variant AC; see Table 5.3. However, with variant FLC more than twice the number of scenarios could be solved to optimality and also the geometric mean time to prove optimality is less than half. Thus, variant FLC seems to be better suited for optimization although AC yields tighter relaxations. However, combining the two variants performs even better.

Remark 5.35. Although the results here show no improvement for determining infeasibility through the usage of $\mathcal{P}^{\mathrm{AS}\pm}$, this is not true in general. It seems that a lot of the scenarios can be identified as infeasible during presolving due to tight pressure bounds in some parts of the network. If we relax the pressure bounds by 1 bar, then more than 3000 scenarios are feasible. If we additionally use a more restrictive model for compressor stations, in particular, the model given by (4.16) *with* the additional facets, then with model variant NFD 2448 scenarios are identified as infeasible and 115 of them are already infeasible after presolving. With model FLC+AC 2637 scenarios are identified as infeasible, 919 of them already during presolving. Note that the 2637 infeasible scenarios of model FLC+AC include the scenarios which are infeasible with model NFD and the difference of 189 scenarios ran into the time limit with model NFD. However, the geometric mean solving times are still quite fast with 13.62 and 4.80 seconds, respectively.

Remark 5.36. The fact that potential-based flows are acyclic does not only hold for this algebraic model for stationary gas flows. It also applies to the stationary

Table 5.5. Aggregated results for `GasLib-582` scenarios for the ODE model.

variant	opt	feas	limit	inf	inf-presol
NFD	41	93	2042	2051	2038
FLC+AC	1784	349	0	2094	2080

Table 5.6. Geometric means of solving times in seconds and total run time in hours for the `GasLib-582` scenarios for the ODE model.

variant	to opt	to first	to inf	total	total time [h]
NFD	864.49	1233.67	1.02	67.46	2149.23
FLC+AC	720.42	73.75	1.02	31.91	829.31

model without height differences based on ODEs which was presented in the previous chapter. To see this, consider a dicycle $(v_1, v_2, \ldots, v_k)$ of pipelines and assume that there is positive flow on each arc (v_i, v_{i+1}) with $i \in [k]$ and (v_k, v_1). Recall that by Corollary 2.17 the pressure decreases in the direction of the flow. Thus, positive flow on each arc in the cycle implies

$$p_{v_1} > p_{v_2} > \ldots > p_{v_k} > p_{v_1}$$

which is a contradiction, i.e., flow in a cycle is not possible for the ODE model, too.

Tables 5.5 and 5.6 display results for the ODE model with the variants NFD and FLC+AC. In particular, the results show an even stronger effect of using FLC+AC. Note that for these computations, we used $\nu_c = 0.4$, the feasibility tolerances 10^{-6} (SCIP default value) for solving the LP-relaxations and $\delta_1 = \delta_2 = 10^{-1}$ for the ODE constraints; see Corollary 4.7. Moreover, the additional bound tightening methods discussed in Section 6.2 were used to strengthen the bounds of the pressure and flow variables.

With model NFD, almost half of the scenarios ran into the time limit without a feasible solution. Only 41 scenarios could be solved to optimality and feasible solutions were found for only 93 further scenarios. In contrast, with model FLC+AC all scenarios were either proven to be infeasible or a feasible solution was found. Moreover, 84 % of the feasible scenarios were solved to optimality.

A more detailed study of computational experiments with the ODE model will be presented in the next chapter. There, we will again consider all model variants from NFD to FLC+AC and study their influence when used together with additional flow tightening techniques.

CHAPTER 6

Implementation and Numerical Results

In this chapter, we finally present details on our implementation of the convex relaxation Algorithm 4.1 and the adaptive spatial branch-and-bound Algorithm 4.2 with the branch-and-bound framework SCIP [40, 132] and computational results for gas network instances from the library `GasLib` [41, 126].

To this end, we show how the models for gas network elements, which were introduced in Section 4.2, are realized with SCIP in the next section. In particular, we will discuss how the implicit trapezoidal rule is solved with a bound preserving adaption of Newton's method; see also Remark 2.20. Then we present problem specific bound tightening techniques in Section 6.2. For pressure and flow variables we have implemented bound tightening techniques based on the lower and upper bounds produced by the explicit midpoint method and the trapezoidal rule; see Section 2.3. Moreover, we have implemented a variant of optimality-based bound tightening (OBBT) for the flow variables. In Section 6.3, the computational setup is introduced, and results for the gas network instance `GasLib-40` and first results for the instance `GasLib-582` are discussed. In Section 4.6, we already presented first results for the network `GasLib-40` and identified the LP-relaxation as well as the presolving as the main reasons for the poor performance. We show that we can successfully overcome these issues by using the combinatorial model $\mathcal{P}^{\mathrm{AS}\pm}$ developed in Chapter 5 and the bound tightening techniques presented in Section 6.2. Subsequently in Section 6.3.1, we discuss numerical issues which we are facing and in Sections 6.3.2 and 6.3.3 we investigate the influence of the objective function and the compressor station model. Finally, in Section 6.3.4 we investigate the performance of our implementation of the convex relaxation Algorithm 4.1 and the adaptive spatial branch-and-bound Algorithm 4.2 for the gas network instance `GasLib-582` similar to Section 5.4. Therefore, we compare results for the model variants introduced in Section 5.4.2 with and without the additional flow tightening techniques. These results show that our methods significantly speed-up the solving process such that we can solve more than 90 % of the 4227 scenarios which are available for the network `GasLib-582` within a time limit of one hour.

6.1 Model Implementation with SCIP

SCIP is a general purpose solver for *solving constraint integer programs* and provides an LP-based branch-and-bound framework for mixed-integer nonlinear programming; see Vigerske and Gleixner [151] for details on the solving process. Besides a variety of general heuristics, branching rules, separators, presolvers, and propagators, SCIP relies on so-called *constraint handlers* for specific types of constraints. These constraint handlers always provide constraint specific feasibility checks and enforcement routines to resolve infeasibility of LP solutions w.r.t. the constraint, e.g., by adding a (linear) constraint to cut off the solution, by reducing the domain of a variable or by branching. Furthermore, they usually contain constraint specific presolving, bound propagation and separation methods. To realize our models presented in Section 4.2 we use the following constraint types which are implemented in SCIP with variables $x \in \mathbb{R}^n$, $y, \tilde{y} \in \mathbb{R}$ and $z \in \mathbb{Z}$, and left-hand sides $l \in \mathbb{R} \cup \{-\infty\}$ and right-hand sides $r \in \mathbb{R} \cup \{\infty\}$:

Linear Linear constraint with $\alpha \in \mathbb{R}^n$:

$$l \;\leq\; \alpha^\top x \;\leq\; r$$

Variable Bound Linear constraint with $\beta \in \mathbb{R}$:

$$l \;\leq\; y + \beta\, z \;\leq\; r$$

Quadratic Nonlinear constraint with $\alpha \in \mathbb{R}^{n\times n}$ and $\beta \in \mathbb{R}^n$:

$$l \;\leq\; \sum_{i,j=1}^{n} \alpha_{i,j}\, x_i\, x_j + \sum_{i=1}^{n} \beta_i\, x_i \;\leq\; r$$

Absolute Power Nonlinear constraint with α, $\beta \in \mathbb{R}$ and $\gamma > 1$:

$$l \;\leq\; \operatorname{sgn}(y+\alpha)|y+\alpha|^\gamma + \beta\, \tilde{y} \;\leq\; r$$

Note that the variables x, y, $\tilde{y}$ in these constraints can be both continuous or integer variables, but the variable z in variable bound constraints has to be binary or integer. Further constraint types are, for example, knapsack constraints, cardinality constraints, or general nonlinear constraints which are represented by using expression trees.

In the following, we present our realization of the models from Section 4.2 and the Weymouth equation (1.9) which we used for the computations in Chapter 5. Since the models of short cuts, valves, control valves and compressors only consist of linear constraints, they can directly be represented by linear and variable bound constraints. Thus, we do not consider their models again, but we present some simplifications for resistors in control valve stations or compressor stations.

6.1.1 Linear Resistors

Our model for linear resistors is given by the constraints (4.9) and (4.10). Since these constraints are linear, they are directly implemented by using the linear constraint type. However, recall that these constraints are a piecewise linear approximation of the discontinuous pressure loss function (4.7)

$$p_u - p_v = \begin{cases} \xi_a & \text{if } q_a > 0, \\ 0 & \text{if } q_a = 0, \\ -\xi_a & \text{if } q_a < 0, \end{cases}$$

with fixed pressure loss $\xi_a > 0$. Since control valves and compressors are one-directional elements, i.e., flow over these elements is nonnegative, and their models already contain binary variables which represent their state, we can directly integrate the pressure loss induced by a linear resistor into the models of control valve stations and compressor stations instead of representing the linear resistor via a separate arc.

Linear Resistors in Control Valve Stations

Consider a control valve station with two linear resistors $r_{in} = (\tilde{u}, u)$, $r_{out} = (v, \tilde{v})$ and the actual control valve $cv = (u, v) \in \mathcal{A}^{cv}$; see Figure 4.1. The pressure variables p_u and p_v have to satisfy the constraints (4.13), i.e.,

$$\begin{aligned} &(\underline{p}_u - \overline{p}_v)(1 - z_{cv}) + \underline{\Delta}_{cv}\, z_{cv} \le p_u - p_v, \\ &p_u - p_v \le (\overline{p}_u - \underline{p}_v)(1 - z_{cv}) + \overline{\Delta}_{cv}\, z_{cv}, \\ &\underline{p}_{cv}\, z_{cv} \le p_u, \\ &p_v \le \overline{p}_{cv}\, z_{cv} + \overline{p}_v(1 - z_{cv}), \end{aligned}$$

where $\underline{\Delta}_{cv}$ and $\overline{\Delta}_{cv}$ denote the minimal and maximal pressure reduction, and z_{cv} is the binary variable representing the state of the control valve; see also Table 4.6. Since the flow on the resistors is equal to the flow on the control valve and only nonzero if $z_{cv} = 1$, we can integrate the pressure loss ξ_{in} and ξ_{out} into these constraints as follows. If the control valve is active, then p_u has to satisfy $p_u \geq \underline{p}_{cv}$ This constraint on p_u is obviously satisfied if $p_{\tilde{u}} \geq \underline{p}_{cv} + \xi_{in}$ holds. Analogously, $p_v \leq \overline{p}_{cv}$ holds if $p_{\tilde{v}} \leq \overline{p}_{cv} - \xi_{out}$ is true. Moreover, the minimal and maximal pressure differences $\underline{\Delta}_{cv}$ and $\overline{\Delta}_{cv}$ between nodes u and v correspond to minimal and maximal pressure differences $\underline{\Delta}_{cv} + \xi_{in} + \xi_{out}$ and $\overline{\Delta}_{cv} + \xi_{in} + \xi_{out}$ between nodes $\tilde{u}$ and $\tilde{v}$. Thus, if we suppose for simplicity that the pressure bounds on p_u and $p_{\tilde{u}}$ as well as p_v and $p_{\tilde{v}}$ are equal, we can combine the models for the resistors and control valve by using the constraints

$$
\begin{aligned}
&(\underline{p}_u - \overline{p}_v)(1 - z_{cv}) + (\underline{\Delta}_{cv} + \xi_{in} + \xi_{out})\, z_{cv} \leq p_{\tilde{u}} - p_{\tilde{v}},\\
&p_{\tilde{u}} - p_{\tilde{v}} \leq (\overline{p}_u - \underline{p}_v)(1 - z_{cv}) + (\overline{\Delta}_{cv} + \xi_{in} + \xi_{out})\, z_{cv},\\
&(\underline{p}_{cv} + \xi_{in})\, z_{cv} \leq p_{\tilde{u}},\\
&p_{\tilde{v}} \leq (\overline{p}_{cv} - \xi_{out})\, z_{cv} + \overline{p}_v(1 - z_{cv}).
\end{aligned}
$$

Linear Resistors in Compressor Stations

Recall that the model for a compressor station $cv = (u, v) \in \mathcal{A}^{cs}$ contains additional pressure variables p_c^{in}, p_c^{out} and a binary variable z_c for every configuration $c \in \mathcal{C}_{cs}^{cf}$ of the compressor station; see also Table 4.7. For each configuration the pressure variables are coupled with p_u and p_v through the constraints

$$
\begin{aligned}
&(\underline{p}_u - \overline{p}_c^{in})\,(1 - z_c) \leq p_u - p_c^{in} \leq (\overline{p}_u - \underline{p}_c^{in})\,(1 - z_c),\\
&(\underline{p}_v - \overline{p}_c^{out})\,(1 - z_c) \leq p_v - p_c^{out} \leq (\overline{p}_v - \underline{p}_c^{out})\,(1 - z_c).
\end{aligned}
$$

Thus, if linear resistors are part of the compressor station, see Figure 4.2, then we can integrate the corresponding pressure loss ξ_{in}, $\xi_{out} > 0$ into these constraints which yields

$$
\begin{aligned}
&(\underline{p}_u - \overline{p}_c^{in})\,(1 - z_c) \leq p_u - p_c^{in} - \xi_{in}\, z_c \leq (\overline{p}_u - \underline{p}_c^{in})\,(1 - z_c),\\
&(\underline{p}_v - \overline{p}_c^{out})\,(1 - z_c) \leq p_v - p_c^{out} + \xi_{out}\, z_c \leq (\overline{p}_v - \underline{p}_c^{out})\,(1 - z_c).
\end{aligned}
$$

6.1.2 Nonlinear Resistors

The pressure loss induced by a nonlinear resistor $a = (u, v) \in \mathcal{A}^{re}$ is given by the nonlinear equation (4.12)

$$
p_u^2 - p_v^2 + |p_u - p_v|(p_u - p_v) = 2\beta_a |q_a| q_a.
$$

To realize this equation with the specific constraint types listed above, we introduce the additional variables π_u, π_v, Δ_a, Δ_a^{abs} and q_a^{abs} representing the squared pressure variables at nodes u and v, the pressure difference $p_u - p_v$, and the signed squares of Δ_a and the flow q_a, respectively. That is, we can represent equation (4.12) by

$$
\begin{aligned}
&\pi_u = p_u^2, \quad \pi_v = p_v^2,\\
&\Delta_a = p_u - p_v, \quad \Delta_a^{abs} = \operatorname{sgn}(\Delta_a)\,\Delta_a^2,\\
&q_a^{abs} = \operatorname{sgn}(q_a)\,q_a^2,\\
&\pi_u - \pi_v + \Delta_a^{abs} - 2\beta_a\,q_a^{abs} = 0.
\end{aligned}
$$

All of these constraints correspond to the constraint types linear, quadratic, and absolute power.

If a nonlinear resistor is part of a compressor station, we know the flow direction in advance – w.l.o.g. let q_a be nonnegative – and thus we can use a simplification of the model above. In this case, we only introduce one additional variable Δ_a, again representing the pressure difference. Since q_a is nonnegative, we do not require the variables for the (signed) squares. Instead we can directly use $0 \le \Delta_a$ and represent equation (4.12) with one linear and one quadratic constraint via

$$
\begin{aligned}
&\Delta_a = p_u - p_v,\\
&p_u^2 - p_v^2 + \Delta_a^2 - 2\beta_a\,q_a^2 = 0.
\end{aligned}
$$

We remark that we could also use this simplification for nonlinear resistors in control valve stations, however, our test instances do not contain such resistors.

6.1.3 Pipelines with Potential-Based Flow Model

For the numerical results in Chapter 5 we used the Weymouth equation (1.9)

$$p_u^2 - p_v^2 = \beta_a\,q_a|q_a|$$

to describe the gas flow through a pipeline $a = (u,v) \in \mathcal{A}^{pi}$. Since this equation is similar to the pressure loss equation of nonlinear resistors, we handle it similarly. We introduce additional variables π_u and π_v representing the squared pressure variables and Δ_a now representing the difference $\pi_u - \pi_v$ which is then coupled to the flow via an absolute power constraint. Altogether we model the Weymouth equation via the constraints

$$
\begin{aligned}
&\pi_u = p_u^2, \quad \pi_v = p_v^2,\\
&\Delta_a = \pi_u - \pi_v,\\
&\beta_a^{-1}\Delta_a = \operatorname{sgn}(q_a)\,q_a^2.
\end{aligned}
$$

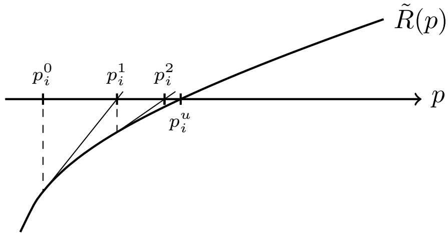


Figure 6.1. Application of Newton's method to solve one step of the trapezoidal rule given by $\tilde{R}(p_i^u) = R(h, p_{i-1}^u, p_i^u, q) = 0$.

6.1.4 Pipelines with ODE Model

To handle the gas flow model based on differential equations we have implemented a *constraint handler* for constraints given by the stationary isothermal Euler equation (1.8) with $\sigma = 0$. At its core this constraint handler provides the functionality of Algorithm 4.1, that is, a feasibility check based on the explicit midpoint method and the trapezoidal rule, and separation routines which produce under- and overestimators as described in Sections 4.3.1 and 4.3.2. Furthermore, the constraint handler contains the bound tightening techniques which will be introduced in the next section and is integrated in the conflict analysis provided by SCIP; see Achterberg [1] for more information. Conflict analysis tries to find cuts to avoid the exploration of infeasible regions of the branch-and-bound tree based on the information of bound changes which led to infeasible subproblems. For a pipeline $a = (u, v)$ with mass flow $q_a \geq 0$ this information is, for example, whether decreasing the upper bound $\overline{p}_u$ led to a new smaller upper bound $\overline{p}_v$.

The basis for all methods and functionality of the constraint handler is given by evaluating the explicit midpoint method (2.18) and trapezoidal rule (2.19), and Lemma 2.23, which shows that the two one-step methods define convex lower and upper bounds P_a^ℓ and P_a^u for every pipeline $a \in \mathcal{A}$. Since the midpoint method is explicit, we can straightforwardly evaluate the method and also compute the derivatives with respect to mass flow and pressure; see Section 4.3.1. However, the trapezoidal rule is an implicit method and we already observed in Remark 2.20 that using standard Newton's method to solve each step of the trapezoidal rule can lead to numerical errors.

Recall from Section 2.3 that for $q \geq 0$ and $\sigma = 0$ the trapezoidal rule applied to ODE (1.8) is given by

$$p_0^u = p^0, \quad p_i^u = p_{i-1}^u - \tfrac{h}{2}\left[\varphi\big(p_{i-1}^u, q\big) + \varphi\big(p_i^u, q\big)\right] \quad \forall i \in [N].$$

Thus, for $i \in [N]$ one step is given by solving the equation

$$R\big(h, p_{i-1}^u, p_i^u, q\big) = p_i^u - p_{i-1}^u + \tfrac{h}{2}\left[\varphi\big(p_{i-1}^u, q\big) + \varphi\big(p_i^u, q\big)\right] = 0.$$

In the following we use $\tilde{R}(p) \coloneqq R(h, p_{i-1}^u, p, q)$. Moreover, suppose that $q > 0$ holds. Otherwise, we have $p_i^u = p^0$ for all $i \in [N]$. Then, if we apply Newton's method to solve this equation for p_i^u with starting point $p_i^0 = p_{i-1}^u$, a new iterate p_i^n is given as a solution of the equation

$$\tilde{R}\big(p_i^{n-1}\big) + \partial_p \tilde{R}\big(p_i^{n-1}\big)\left(p_i^n - p_i^{n-1}\right) = 0$$

for all $n \in \mathbb{N}$. Since $\tilde{R}(p)$ is increasing and concave in p, this implies $\tilde{R}(p_i^n) < 0$ and thus $p_i^n < p_i^u$ for all $n \in \mathbb{N}$. That is, Newton's method produces lower bounds on p_i^u; see Figure 6.1. Since this error increases by solving N steps of the trapezoidal rule, this can actually lead to solutions $p_N^u < p_N^\ell$, that is, what should be an upper bound on the solution is a lower bound.

To resolve this issue, observe that by choosing $h = \frac{L}{N}$ according to Lemma 2.23 we get $h\, \partial_p \varphi(p, q) \le 1$ and thus

$$1 < \partial_p \tilde{R}(p) = 1 + \tfrac{h}{2}\, \partial_p \varphi(p, q) \le \tfrac{3}{2}. \tag{6.1}$$

Now, suppose that $\tilde{R}(p_i^{n-1}) > 0$ holds, i.e., $p_i^{n-1} > p_i^u$ and let

$$p_i^n = p_i^{n-1} - \frac{2}{3}\,\frac{\tilde{R}(p_i^{n-1})}{\partial_p \tilde{R}(p_i^{n-1})},$$

that is, we define p_i^n by using two-thirds of the regular Newton step length. Then using that $\tilde{R}$ is concave yields

$$\begin{aligned}
\tilde{R}(p_i^n) &\ge \tilde{R}(p_i^{n-1}) + \partial_p \tilde{R}(p_i^n)\left(p_i^n - p_i^{n-1}\right) \\
&= \tilde{R}(p_i^{n-1}) - \frac{2}{3}\,\frac{\partial_p \tilde{R}(p_i^n)}{\partial_p \tilde{R}(p_i^{n-1})}\,\tilde{R}(p_i^{n-1}) \\
&> \tilde{R}(p_i^{n-1}) - \frac{2}{3}\,\frac{3}{2}\,\tilde{R}(p_i^{n-1}) = 0.
\end{aligned}$$

Therefore, p_i^n is an upper bound on p_i^u. We can exploit this fact to derive a bound preserving Newton method for evaluating the trapezoidal rule as described in Algorithm 6.1.

We initialize Algorithm 6.1 with the current iterate p_{i-1}^u of the trapezoidal rule, the step size h, and two tolerances $\varepsilon_1, \varepsilon_2 > 0$. In the first phase of the algorithm, we perform the standard Newton's method with starting point $p_i^0 = p_{i-1}^u$ to compute a

Algorithm 6.1 Bound preserving Newton method for evaluating (2.19)

Input: Starting pressure p_{i-1}^u, step size h, tolerances ε_1, $\varepsilon_2 > 0$.

Output: Upper bound p_i^n on the solution p_i^u of $R(h, p_{i-1}^u, p_i^u, q) = 0$ with $\tilde{R}(p_i^n) \leq \varepsilon_2$.

1: Let $p_i^0 \leftarrow p_{i-1}^u$ and $n \leftarrow 0$.
2: **While** $\tilde{R}(p_i^n) < -\varepsilon_1$ **do**
3: Let $n \leftarrow n + 1$.
4: Set $p_i^n \leftarrow p_i^{n-1} - \frac{\tilde{R}(p_i^{n-1})}{\partial_p \tilde{R}(p_i^{n-1})}$.
5: **While** $\tilde{R}(p_i^n) < 0$ **do**
6: Set $p_i^n \leftarrow p_i^n - \frac{\tilde{R}(p_i^{n-1})}{\partial_p \tilde{R}(p_i^{n-1})}$.
7: **While** $\tilde{R}(p_i^n) > \varepsilon_2$ **do**
8: Let $n \leftarrow n + 1$.
9: Set $p_i^n \leftarrow p_i^{n-1} - \frac{2}{3} \frac{\tilde{R}(p_i^{n-1})}{\partial_p \tilde{R}(p_i^{n-1})}$.
10: **return** p_i^n.

solution p_i^n which is already close to the exact solution of $\tilde{R}(p) = 0$, but still a lower bound. For that, we use the tolerance ε_1 and the stopping criterion $\tilde{R}(p_i^n) \geq -\varepsilon_1$. Next, we iteratively add the last step length to p_i^n until it satisfies $\tilde{R}(p_i^n) \geq 0$, i.e., until p_i^n is an upper bound on p_i^u. Afterwards, we apply Newton's method again, but with only two-thirds of the regular step length, until we find a solution p_i^n which satisfies $\tilde{R}(p_i^n) < \varepsilon_2$. By the arguments above, this is an upper bound on p_i^u.

By default we use the tolerances $\varepsilon_1 = 0.5\,\text{Pa}$ and $\varepsilon_2 = 10^{-5}\,\text{Pa}$ for Algorithm 6.1. Furthermore, if we apply the trapezoidal rule in the direction of the flow to compute lower bounds on the output pressure, see Remark 2.22, we use the same adaption of Newton's method. We then additionally check if there is a solution $p_i^n \geq \frac{c_{\prime} q}{\nu_c A}$. Moreover, we use one-half instead of two-thirds of the regular step length in the third phase of the algorithm for the following reason. Since the trapezoidal rule in the direction of the flow is given by

$$R\big(h, p_{i-1}^u, p_i^u, q\big) = p_i^u - p_{i-1}^u - \tfrac{h}{2}\left[\varphi\big(p_{i-1}^u, q\big) + \varphi\big(p_i^u, q\big)\right] = 0$$

with the pressure variables minus (instead of plus) the evaluations of φ. Analog to equation (6.1) we get that $\partial_p \tilde{R}(p)$ is bounded, however, then instead by

$$1 > \partial_p \tilde{R}(p) = 1 - \tfrac{h}{2}\, \partial_p \varphi(p, q) \geq \tfrac{1}{2}.$$

6.2 Bound Tightening Techniques

In (spatial) branch-and-bound approaches for mixed-integer nonlinear optimization bound tightening techniques are typically applied during presolving and in every node of the branch-and-bound tree; for example, see Belotti et al. [11] for a description of bound tightening techniques used in the solver COUENNE. While from a theoretical point of view bound tightening is not necessary to ensure convergence, it has a big impact on the performance in practice as shown in the recent survey by Puranik and Sahinidis [112]. In particular, BARON [123] is named after the branch-and-reduce algorithm by Ryoo and Sahinidis [118], where the "reduce" stands for domain reduction of the variables. Since bound tightening is important for the performance, we have implemented problem specific methods to be used alongside the techniques already implemented in SCIP.

Bound tightening techniques can roughly be categorized as *feasibility-based* and *optimality-based*. As an example for feasibility-based bound tightening, consider a linear inequality constraint $\sum_{i=1}^{d} \alpha_i x_i \leq \beta$ and variable bounds $\underline{x} \leq x \leq \overline{x}$. Then for x_j with $\alpha_j \neq 0$ we can use

$$\alpha_j\, x_j \leq \beta - \sum_{i \neq j,} \alpha_i\, x_i \leq \beta - \sum_{i \neq j,\, \alpha_i > 0} \alpha_i\, \underline{x}_i - \sum_{i \neq j,\, \alpha_i < 0} \alpha_i\, \overline{x}_i$$

to possibly derive a new upper bound if $\alpha_j > 0$ or lower bound if $\alpha_j < 0$. This procedure is also known as *bound propagation* and can be applied to nonlinear constraints if they are nondecreasing or nonincreasing in the variables. The idea of *optimality-based bound tightening* (OBBT) is to minimize or maximize a variable, typically over a relaxation of the original problem, to derive new bounds. For enhancements of this basic idea see Gleixner et al. [47].

We will now present bound tightening of pressure variables based on bound propagation using the functions P^ℓ and P^u; see Lemma 2.23. Afterwards, we introduce bound tightening for flow bounds of pipes based on bound propagation and bisection. Furthermore, we discuss our implementation of OBBT for general flow bounds.

6.2.1 Pressure Bounds

Consider a pipe $a = (u, v) \in \mathcal{A}^{pi}$ with pressure variables $\underline{p}_u \leq p_u \leq \overline{p}_u$, $\underline{p}_v \leq p_v \leq \overline{p}_v$ and mass flow $\underline{q}_a \leq q_a \leq \overline{q}_a$. By Lemma 2.23 we know that P^ℓ and P^u are nondecreasing and define lower and upper bounds on the input pressure w.r.t. the flow direction. We can exploit these properties to perform bound propagation as follows. To this end, we consider the two cases $0 \leq \underline{q}_a$ and $\underline{q}_a < 0 < \overline{q}_a$. Note that

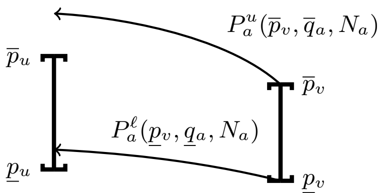


Figure 6.2. Example for bound propagation of pressure bounds with the functions P_a^ℓ and P_a^u in the case $0 < \underline{q}_a$.

the case $\overline{q}_a \leq 0$ can be treated analogously to the first case by interchanging the pressure variables, and using the mass flow $-q_a$.

Case $0 \leq \underline{q}_a$**.** In this case, the inequality $P_a^\ell(p_v, q_a, N_a) \leq p_u \leq P_a^u(p_v, q_a, N_a)$ holds. Thus, by using monotonicity we get

$$P_a^\ell(\underline{p}_v, \underline{q}_a, N_a) \leq p_u \leq P_a^u(\overline{p}_v, \overline{q}_a, N_a).$$

Hence, if $\underline{p}_u < P_a^\ell(\underline{p}_v, \underline{q}_a, N_a)$ or $P_a^u(\overline{p}_v, \overline{q}_a, N_a) < \overline{p}_u$, we can tighten the bounds of pressure variable p_u. Figure 6.2 shows an example, where we can derive a new lower bound. Moreover, we can use this to detect infeasibility of the (sub)problem, e.g., if inequality $\overline{p}_u < P_a^\ell(\underline{p}_v, \underline{q}_a, N_a)$ holds, there is no feasible solution.

Case $\underline{q}_a < 0 < \overline{q}_a$**.** In the second case, we can still compute an upper bound on the variable p_u through $P_a^u(\overline{p}_v, \overline{q}_a, N_a)$. But due to $\underline{q}_a < 0$ we cannot derive a new lower bound on either of the pressure variables with P^ℓ and P^u. Instead, we can propagate the upper bound on p_u with the lower flow bound and try to compute a new upper bound on p_v, that is, an upper bound on p_v is given by

$$p_v \leq P_a^u(\overline{p}_u, -\underline{q}_a, N_a).$$

Furthermore, it is also possible to detect infeasibility in this case.

Recall from Remarks 2.5 and 2.22 in Chapter 2 that we can apply the explicit midpoint method (2.18) and the trapezoidal rule (2.19) in the direction of the flow as well, i.e., start with the input pressure as initial value instead of the output pressure. If we can successfully evaluate both methods, that is, $0 \leq \nu_c A \, p_N^{\ell/u} - c \, q$ holds, then the trapezoidal rule defines a lower bound on the output pressure and the explicit midpoint method defines an upper bound. We use this for bound propagation as well. Therefore, denote with $P^{\ell,+}(p, q, N)$ and $P^{u,+}(p, q, N)$ the evaluation of the two methods in the direction of the flow.

Case $0 \leq \underline{q}_a$**.** The output pressure is increasing with respect to the input pressure and decreasing with respect to the mass flow. Hence, we can try to compute new bounds on p_v via

$$P_a^{\ell,+}(\underline{p}_u, \overline{q}_a, N_a) \le p_v \le P_a^{u,+}(\overline{p}_u, \underline{q}_a, N_a).$$

Case $\underline{q}_a < 0 < \overline{q}_a$. Analogously to computing bounds in the reverse direction of the flow, propagating $\overline{q}_a$ with $P_a^{\ell,+}(\underline{p}_u, \overline{q}_a, N_a)$ is still a valid lower bound on p_v. Moreover, a new lower bound on p_u can possibly be computed via

$$P_a^{\ell,+}(\overline{p}_v, -\underline{q}_a, N_a) \le p_u.$$

Note that in both cases it is again possible to detect infeasibility.

We repeatedly apply bound propagation during presolving and in every node of the branch-and-bound tree before the LP-relaxation is solved. To avoid unnecessary evaluations of the numerical schemes, we only propagate bounds if at least one bound has changed since we last tried to derive a new bound. That is, for example in the first case, we only check whether $P_a^u(\overline{p}_v, \overline{q}_a, N_a)$ defines a tighter bound on p_u if at least one of the variable bounds $\overline{p}_v$ or $\overline{q}_a$ has changed due to branching or further bound tightening.

Note that bound propagation for pressure has been used for all computations with the ODE model in this thesis. It is quite fundamental for the performance of our implementation. Thus, we did not implement an option to turn it on or off and did not test the influence of bound propagation on solving capabilities and running times.

6.2.2 Flow Bounds

In Section 4.6, we observed that the flow bounds often are still very large after presolving, which motivated not only the study of the combinatorial models in Chapter 5, but also the implementation of flow tightening techniques based on bound propagation and optimality-based bound tightening. But first of all notice the following simple test whether the flow direction of a pipe $a = (u, v) \in \mathcal{A}^{pi}$ can be fixed. Whenever we perform bound propagation for the pressure bounds, we also check if either $\underline{p}_u \ge \overline{p}_v$ or $\overline{p}_u \le \underline{p}_v$ holds. Since we assumed $\sigma = 0$, the pressure decreases in the direction of the flow, and thus the flow has to be nonnegative in the first case or nonpositive in the second case.

Note that we have implemented the flow tightening techniques presented in the following only for the ODE model, i.e., we did not use them for the computations with the algebraic model in Chapter 5. Furthermore, both methods are optional, that is, they can be turned on or off, and in particular we study their influence on the performance of our spatial branch-and-bound algorithm in Section 6.3.4.

Bound Propagation

Consider a pipe $a = (u, v) \in \mathcal{A}^{pi}$ with $0 \leq \underline{q}_a$. Our bound propagation method for flow bounds is based on the observation that if

$$\overline{p}_u < P_a^\ell(\underline{p}_v, \overline{q}_a, N_a)$$

holds, there is no feasible solution with $q_a = \overline{q}_a$. We then apply bisection to the flow interval $[\underline{q}_a, \overline{q}_a]$ to find a new upper bound on the flow. We stop the bisection procedure if we find a flow value $\tilde{q}_a$ such that $P_a^\ell(\underline{p}_v, \tilde{q}_a, N_a)$ satisfies

$$\overline{p}_u \leq P_a^\ell(\underline{p}_v, \tilde{q}_a, N_a) \leq \overline{p}_u + 1.0\,\text{bar}.$$

Note that we use the confidence interval of 1.0 bar to avoid long running times of the bisection procedure. Moreover, note that other approaches than bisection are possible to find a new upper flow bound. For example, we tested variants of bisection which use gradient information of P_a^ℓ to determine a point for dividing the interval other than using the midpoint. However, it turned out that standard bisection is faster. Furthermore, if the lower flow bound satisfies $\underline{q}_a < 0$, we apply bisection to the interval $[0, \overline{q}_a]$ instead and we can also use this idea to increase the lower flow bound if we have

$$\overline{p}_v < P_a^\ell(\underline{p}_u, -\underline{q}_a, N_a).$$

We apply bound propagation for flow variables immediately before we apply bound propagation to the pressure variables, i.e., during presolving as well as in every node of the branch-and-bound tree. Just like before, we only try to compute new bounds on the flows, if at least one of the pressure bounds of p_u and p_v has changed.

Optimality-Based Bound Tightening

In contrast to the bound propagation method described above, we can apply OBBT to all network elements. Since minimizing or maximizing a variable over the original constraints would be as hard as the problem we want to solve itself, we only minimize and maximize flow variables over a mixed-integer linear relaxation. The particular relaxation $\mathcal{R}$, which we solve, is given by removing all nonlinear constraints from the original problem (4.18), i.e., we remove all quadratic and absolute power constraints, see Section 6.1, and all ODE constraints. Moreover, the combinatorial model variants presented in Chapter 5 are used in the relaxation for OBBT as well if they are included in the current problem. To derive new bounds on a flow variable q_a for arc $a \in \mathcal{A}$, we then solve

$$\min / \max \quad q_a \qquad \text{s.t. } (p, q, z) \in \mathcal{R}.$$

Since solving two optimization problems for every flow variable can be very time consuming, we take the following measures:

- We perform OBBT only once at the end of presolving.
- We try to compute new bounds on variables only if they are not already “close.” For all computations in this thesis we only performed OBBT on flow variables q_a with
$$\overline{q}_a - \underline{q}_a > 10.0\,\mathrm{kg\,s^{-1}}.$$
- In total we use a time limit of 10 minutes for OBBT.
- We use reoptimization, i.e., if we change the objective from minimizing to maximizing or from one variable to another, we do not solve the problem from scratch. Instead, we use previously found feasible solutions to speed-up the solution process.

6.3 Computational Experiments

In this section, we present computational results to demonstrate the capabilities of the adaptive spatial branch-and-bound Algorithm 4.2. Therefore, we first present the computational setup and come back to the earlier results for the gas network `GasLib-40` and the problems discussed in Section 4.6. We will see that by using the combinatorial model $\mathcal{P}^{\mathrm{AS}\pm}$ from Chapter 5 and the flow tightening techniques introduced above, we can successfully improve the performance of the presolving and the running times. Afterwards, we present computations for the larger gas network `GasLib-582`. We discuss some numerical issues, which we are facing, the influence of the objective function and the compressor station model. In the end, we present a detailed study of numerical results similar to Section 5.4, where we presented results for the potential-based flow model with the model variants NFD to FLC+AC.

All computations presented in this thesis were performed on a cluster with 3.5 GHz Intel Xeon E5-1620 Quad-Core CPUs, having 32 GB main memory and 10 MB cache running Linux. We used SCIP version `7.0.0` [40, 132] with a time limit of one hour and we used either CPLEX version `12.10.0` or SOPLEX version `5.0.0` as LP-solver.

If not explicitly stated otherwise, we used the following default settings. LP-relaxations were solved with CPLEX and a feasibility tolerance of 10^{-6} (SCIP default value). Moreover, we used the parameter $\nu_c = 0.4$, chose the initial discretizations according to Lemma 2.23, and used tolerances $\delta_1 = \delta_2 = 10^{-1}$ for testing δ-feasibility of the ODE constraints, see Corollary 4.7. The default objective function was to maximize the sum of the pressure variables at the nodes and we used the compressor station model (4.15) by Hiller and coworkers [69, 154] with the additional facets.

Furthermore, by default we used model variant FLC+AC and the flow tightening techniques based on bound propagation and OBBT.

Our test networks `GasLib-40` and `GasLib-582` are part of the library `GasLib` [41, 126] for gas network instances. Particular networks from `GasLib` have also been used in other publications, e.g., see Pfetsch et al. [111], Koch et al. [82], Gugat et al. [53], Schmidt et al. [128], Becker and Hiller [8], or Burlacu et al. [20].

In Section 4.6, we presented first computational results for the network `GasLib-40`, which has 40 nodes, 39 pipes, and 6 compressor stations, for a load scenario with 3 entries and 29 exits. There, we used the parameters $\nu_c = 0.8$ and $\delta_1 = \delta_2 = 10^{-4}$, and the basic model presented in Section 4.2 without binary variables representing the flow directions, i.e., the model NFD. We have seen that for this model presolving has almost no positive effect on the variable bounds of the flow variables. However, if we use the same parameters with the combinatorial model $\mathcal{P}^{\mathrm{AS}\pm}$, i.e., model variant FLC+AC, and the flow tightening techniques introduced in Section 6.2, then presolving can fix three additional flow variables and drastically tightens the arithmetic mean lower and upper bound of the remaining flow variables from -2125.83 kg/s and 2139.25 kg/s to -42.32 kg/s and 87.90 kg/s. Analogously to Figure 5.7, we compare the state of the network after presolving for model variant NFD and for model variant FLC+AC with the flow tightening techniques in Figure 6.3. The figure shows that presolving for model FLC+AC not only fixes three additional flow variables, but also the flow direction on 19 pipelines. The flow direction remains unknown for only four of 39 pipelines. Furthermore, using model FLC+AC leads to a faster running time of 449.24 seconds compared to 972.90 seconds for model variant NFD.

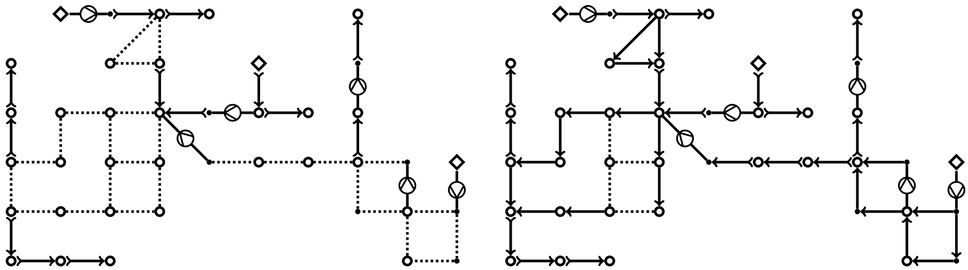


Figure 6.3. The presolved network `GasLib-40` corresponding to the ODE model with variants NFD without flow tightening techniques (left) and FLC+AC with flow tightening (right). The scenario has 3 sources (diamonds) and 29 sinks (circles). Pipes with fixed flow are depicted by $\rightarrowtail$, fixed flow directions are shown by $\rightarrow$, and the remaining pipes (with unfixed flows/directions) are dashed.

Further speed-up of the solving times for instance `GasLib-40` can be obtained by using smaller values for the parameter ν_c and larger feasibility tolerances $\delta_1 = \delta_2$. For example, if we use parameter $\nu_c = 0.4$ and tolerances $\delta_1 = \delta_2 = 10^{-2}$ instead of $\nu_c = 0.8$ and $\delta_1 = \delta_2 = 10^{-4}$, then we can solve the scenario for network `GasLib-40` in 1.13 seconds. Note that using larger feasibility tolerances has only a small effect on the optimal solution of this instance. The optimal value (sum of the pressure variables) changes by less than one bar which is a change of roughly 0.027%. Moreover, $\frac{cq}{Ap} < \nu_c = 0.4$ is satisfied in both optimal solutions.

Since our algorithm is significantly faster with model FLC+AC, we can solve larger gas networks. In particular, for the subsequent computational experiments we use the network `GasLib-582` which has 582 nodes, 278 pipes, 5 compressor stations, 23 control valves, 8 resistors, 26 valves, and 269 short pipes. In total there are 4227 different load scenarios, i.e., inflows and outflows, for this network. These scenarios have been generated by sampling different distributions of loads for different temperature classes. The distributions are based on historical data, which are available (but not publicly) since `GasLib-582` is a distortion of a real-world network in Germany. For details on the scenario generation see Hayn et al. [65].

Remark 6.1. In our default setting we use the parameter $\nu_c = 0.4$ and the feasibility tolerances $\delta_1 = \delta_2 = 10^{-1}$. Using $\nu_c = 0.4$ is not very restrictive and evaluating the fraction $\frac{cq}{Ap}$ in feasible solutions which we find with our algorithm shows that we could also use smaller values for ν_c, e.g., $\nu_c = 0.2$. Since ν_c directly influences the maximal step sizes in the discretizations of the ODE constraints, this would make our algorithm faster; see Lemma 2.23. On the other hand, using a larger value such as $\nu_c = 0.8$ makes the algorithm significantly slower. Moreover, we point out that we can also use smaller feasibility tolerances. Using $\delta_1 = \delta_2 = 10^{-2}$ has only a minor effect on the number of infeasible scenarios for the network `GasLib-582`. With a tolerance of 10^{-1} there are 3134 infeasible scenarios, see Table 6.3 with LP-solver CPLEX, and with tolerance 10^{-2} there are 3140 infeasible scenarios. However, using tolerance 10^{-2} significantly increases the total time for the computation from 417 hours to almost 900 hours.

Remark 6.2. The discretizations used to compute the explicit midpoint method (2.18) and the implicit trapezoidal rule (2.19) mainly depend on the choice of the parameter ν_c and the feasibility tolerances δ_1, δ_2. In fact, the initial discretization only depends on ν_c and the pipelines parameters, i.e., length, diameter and friction coefficient. The smaller ν_c is the larger the step sizes can be chosen. For the network `GasLib-40` using the parameter $\nu_c = 0.2$ leads to initial step sizes between 877.69 m and 3418.01 m, using $\nu_c = 0.4$ to step sizes between 156.03 m and 569.67 m, and $\nu_c = 0.8$ to step sizes between 5.12 m and 19.2 m. For the network

Table 6.1. Results for network `GasLib-582` using the default setting categorized in temperature the classes warm, mild, cool, cold and freezing.

temperature class	opt	feas	limit	inf	inf-presol	total
warm	271	45	2	65	0	383
mild	285	47	6	309	17	647
cool	212	37	8	943	553	1200
cold	149	27	3	933	847	1112
freezing	1	0	0	884	883	885
total	918	156	19	3134	2300	4227

Table 6.2. Geometric mean solving times in seconds and total run time in hours for the results presented in Table 6.1.

temperature class	to opt	to first	to inf	total	total time [h]
warm	574.54	109.43	34.89	447.35	111.83
mild	630.29	97.29	28.69	166.24	131.93
cool	677.15	105.25	4.31	13.56	110.44
cold	501.42	108.36	1.38	3.75	63.03
freezing	190.80	45.60	1.00	1.01	0.09
total	600.03	104.32	2.56	11.31	417.32

`GasLib-582` the same parameters ν_c correspond to initial step sizes between 151.02 m and 4742.95 m, 27.04 m and 822.54 m, and 0.89 m and 26.87 m. Note that using a first-discretize-then-optimize approach as discussed in Section 1.2 with these initial discretizations would require 570, 3486, or 107947 additional pressure variables for network `GasLib-40` and 1533, 9822 or even 306448 additional pressure variables for network `GasLib-582`.

The development of the discretizations during the course of the branch-and-bound process then depends on the feasibility tolerances and the solutions of the relaxations which are found. Of course, using smaller tolerances makes it more likely that the discretization is refined. However, since the the initial step sizes can be rather small, in particular if we use $\nu_c = 0.8$, it can happen that the discretizations are not refined even when using $\delta_1 = \delta_2 = 10^{-4}$. For example, this occurred for instance `mild_2480` of network `GasLib-582` with $\nu_c = 0.8$ and $\delta_1 = \delta_2 = 10^{-4}$. On the other hand, when using the coarse initial discretization resulting from $\nu_c = 0.2$ instead for this instance the smallest step size after the one hour time limit was 9.94 m, while the largest step size still was 4742.95 m.

With our default settings we can solve almost all of the 4227 scenarios for the network `GasLib-582` within the time limit of one hour. Tables 6.1 and 6.2 show results and running times for these scenarios categorized by the temperature classes warm, mild, cool, cold and freezing. Similar to Section 5.4, in Table 6.1 column "opt" displays the number of feasible scenarios solved to optimality, column "feas" the number of scenarios for which a feasible solution was found, but could not be solved to optimality, column "limit" the number of scenarios running into the time limit without a feasible solution, column "inf" the total number of scenarios which have been proven to be infeasible, and "inf-presol" the number of infeasible scenarios where infeasibility has been detected during presolving. Moreover, Table 6.4 shows geometric mean times in seconds and the total running time in hours for these computations. Here, column "to opt" shows the geometric mean time to prove optimality, "to first" the geometric mean time until the first feasible solution was found, "to inf" the geometric mean time to prove infeasibility, "total" the geometric mean time over all scenarios and "total time" shows the total running time of the computations in hours. The results show that for only 19 of 4227 scenarios our algorithm cannot find a feasible solution or prove that the scenario is infeasible. Moreover, in total only 175 scenarios ran into the time limit, whereas the other scenarios can be solved to optimality with a geometric mean time of about 10 minutes or proven to be infeasible with a geometric mean time of less than three seconds.

6.3.1 Numerical Issues

The optimization problem (4.18) with the binary variables for the flow directions and model variant FLC+AC for a scenario of the network `GasLib-582` contains 2579 variables and initially consists of nearly 6300 constraints including almost 6000 linear constraints and 278 ODE constraints. Unfortunately, the LP-relaxations of this problem (in fact for all model variants) are sometimes ill-conditioned. Therefore, we have to be aware of computational accuracy when reading and evaluating the computational results in this chapter. Using pressure variables in bar instead of pascal has the advantage that most pressure bounds are within 1 and 90 instead of 10^5 and $90 \cdot 10^5$. Moreover, then pressure and mass flow variables are within a similar range after presolving; see Table 6.12 for the mean flow bounds on pipes after presolving. The negative side effect of this is that we have to include the conversion factor of 10^5 in the constraints, which can lead to badly scaled inequalities. Consider for example the inequalities

$$
\begin{aligned}
p_{in} - 0.999662423562658\, p_{out} + 0.000214179735239\, q &\leq 0.053861944508877,\\
p_{in} - 0.996842741770021\, p_{out} - 0.001480746986198\, q &\geq 0.001454351582613,
\end{aligned}
$$

Table 6.3. Comparison of results for GasLib-582 scenarios with model variant FLC+AC for LP-solvers CPLEX and SOPLEX.

LP-solver	opt	feas	limit	inf	inf-presol
CPLEX	918	156	19	3134	2300
SOPLEX	938	123	15	3151	2301

Table 6.4. Geometric mean solving times in seconds and total run time in hours for the GasLib-582 scenarios with model variant FLC+AC for LP-solvers CPLEX and SOPLEX.

LP-solver	to opt	to first	to inf	total	total time [h]
CPLEX	600.03	104.32	2.56	11.31	417.32
SOPLEX	548.67	122.20	2.71	11.14	364.13

and

$$157079.6326795\, p_{in} + 1719.10820854487\, q \geq 0$$

which are examples for an underestimator of P^ℓ, an overestimator of P^u, and inequality (4.3) with $\nu_c = 0.4$ produced during one solving process of scenario warm_414. Furthermore, the conversion factor might lead to cancellation and loss of significance during the evaluation of (2.18) and (2.19), when computing the lower and upper bounding functions P^ℓ and P^u.

One consequence of the badly scaled inequalities is that both LP-solvers we use, that is, CPLEX and SOPLEX, infrequently report numerical troubles. While in most cases the LP-relaxations can be solved with tighter feasibility tolerances, there are very seldom cases where the LP-relaxation cannot be solved at all. Another problem which we can observe more often, is that we get contradicting results for the same scenario, when varying the model variant or the LP-solver. That is, sometimes one variant or solver claims to have found a feasible or optimal solution and another variant or solver declares the scenario infeasible. For example, consider Table 6.3 which shows results for all 4227 scenarios for the network GasLib-582 with the LP-solvers CPLEX and SOPLEX. For these computations the model variant FLC+AC and the additional compressor facets were used and the objective was to maximize the sum of the pressures at the nodes. Analogously to before, Table 6.3 displays the number of feasible scenarios solved to optimality, the number of scenarios which ran into the time limit with and without a feasible solution, the number of scenarios proven to be infeasible, and the number of scenarios proven to be infeasible already during presolving. Moreover, Table 6.4 shows the corresponding geometric mean

Table 6.5. Detailed comparison of solving status for LP-solvers CPLEX and SOPLEX. The columns denote the solving status with CPLEX (cpx) and the rows the status with SOPLEX (spx).

spx\cpx	opt	feas	limit	inf
opt	827	99	10	2
feas	73	48	1	1
limit	4	4	7	0
inf	14	5	1	3131

times to optimality, until the first feasible solution was found, to infeasibility, and the geometric mean time over all scenarios in seconds and the total running time in hours.

While at first glance the numbers in Tables 6.3 and 6.4 are very similar and only suggest that SOPLEX is a bit faster, there are 22 scenarios with contradicting results. In Table 6.5 we can see a partitioning of the scenarios by their results with the two LP-solvers. The columns denote the result using CPLEX and the rows denote the result using SOPLEX, e.g., column "feas", row "inf" shows the number of scenarios for which a feasible solution was found using CPLEX, but were infeasible using SOPLEX. Moreover, when comparing the objective values for the scenarios, which are feasible or optimal with both solvers, we found 30 scenarios where one solution value violates the dual bound of the other solver by more than 0.1%. The maximal gap between the corresponding primal and dual bounds we found is 1.861%. Note that the number of contradictions increaseas if we use settings such that more scenarios are feasible. For example, if we do not use the additional facets for the compressor model, then there are 55 scenarios with contradicting results.

Another explanation for contradicting results is the fact that convex underestimators and concave overestimators of nonlinear functions are typically tight in some points and thus cut off δ-feasible solutions. However, when using another model or solver, the solving process usually differs so that slightly other under- and overestimators are generated, which then cut off different δ-feasible solutions. While some of the contradicting results might be due to this fact, it seems very likely that some of the contradictions can be traced back to numerical errors.

Although Table 6.4 shows that SOPLEX is faster than CPLEX, we use CPLEX for the following tests. The reason for this is that in our computational experience CPLEX seems to be more robust in handling numerical problems.

6.3.2 Influence of the Objective Function

Concerning the planning or operation of a gas network there are several questions one can ask. For example, for the long-term planning of network extension or if transport contracts have to be made, the transmission system operator has to know whether certain (future) load scenarios can be realized. Moreover, for the operation itself energy efficient transport is desirable. Thus, there are several objective functions which come into question for us; see also Section 4.2. Nevertheless, we chose as objective function to maximize the sum of the pressure variables at the nodes for demonstrating the effect of using the combinatorial model $\mathcal{P}^{\mathrm{AS}\pm}$ in Section 5.4.2. Furthermore, we will this objective in the subsequent sections as well, even though it has not an interpretation as clear as, for example, finding an energy efficient solution.

To show that this objective function is suitable for demonstrating the capabilities of our algorithm, the effect of using the combinatorial model $\mathcal{P}^{\mathrm{AS}\pm}$ and the flow tightening techniques, we performed tests with the following objective functions:

- Solve problem (4.18) as a pure feasibility problem, i.e., $C(p, q, z) \equiv 0$.
- Minimize the number of running compressors, i.e., $C(p, q, z) = \sum_{cs \in \mathcal{A}} z_{cs}$.
- Minimize the power loss due to transportation, i.e., we use the objective function $C(p, q, z) = \sum_{v \in \mathcal{V}} q_v^{\pm} p_v$ as a proxy for the power required for transportation; see Section 4.2.7.
- Maximize the sum of the pressure variables, i.e., $C(p, q, z) = -\sum_{v \in \mathcal{V}} p_v$.

Since the energy required for the transport is mainly consumend in compressor stations, it would be of particular interest to minimize their power consumption. However, as mentioned before this is not possible with the (current) compressor station model which we are using. Instead, we can minimize the number of running compressors as a substitute.

The results for the network `GasLib-582` are summarized in Tables 6.6 and 6.7, where the columns denote the same as before, i.e., the numbers of optimally solved scenarios, scenarios that ran into the time limit with or without a feasible solution, infeasible scenarios, and the number infeasible scenarios which were already identified as infeasible during presolving in Table 6.6. The geometric mean solving times and the total running time corresponding to these results are shown in Table 6.7.

The numbers show that (for most scenarios) determining infeasibility is easy with each objective function. In total 3136 different scenarios were identified as infeasible and 2300 of them already during presolving. Moreover, for 2 of the 3136 scenarios we have contradicting results, that is, we found a feasible solution for scenario `mild_593` twice and a feasible solution for scenario `mild_3309` once, otherwise these two scenarios have been proven to be infeasible. Furthermore, the geometric mean times to infeasibility are quite fast. The geometric mean time to infeasibility over

Table 6.6. Comparison of results for `GasLib-582` scenarios with different objective functions[1].

objective	opt	feas	limit	inf	inf-presol
min 0	979	0	115	3133	2300
min $\sum z_{cs}$	1023	1	70	3133	2300
min $\sum q_u^{\pm} p_u$	60	868	161	3136	2300
max $\sum p_u$	918	156	19	3134	2300

Table 6.7. Geometric mean solving times in seconds and total run time in hours for the `GasLib-582` scenarios with different objective functions.

objective	to opt	to first	to inf	total	total time [h]
min 0	76.95	76.94	2.54	6.82	165.10
min $\sum z_{cs}$	75.76	75.56	2.54	6.52	122.48
min $\sum q_u^{\pm} p_u$	1556.26	179.08	2.57	16.46	1072.01
max $\sum p_u$	600.03	104.32	2.56	11.31	417.32

all objective functions is less than three seconds. And further, the geometric mean time to infeasibility for the scenarios which are not solved during presolving is still less than 100 seconds for all objective functions.

In contrast to that, the objective functions lead to a quite different behavior of the solution process for the feasible scenarios. We can see that solving the feasibility problem behaves very similar to minimizing the number of running compressors and minimizing the power loss sticks out with more than a quarter of the scenarios running into the time limit. When solving the feasibility problem, we are done if a feasible solution is found. For the objective $C(p, q, z) = \sum_{cs \in \mathcal{A}} z_{cs}$ the times to the first solution and the time to optimality are very close. One possible explanation for this is as follows. Since there are only five compressor stations, there are only six possible results, i.e., $\sum_{cs \in \mathcal{A}} z_{cs} \in \{0, 1, \ldots, 5\}$. Thus, if a feasible solution with no active compressor station is found, then we are done. Otherwise, if a feasible solution with only one or two active compressor stations is found, then it remains to show that the problem is infeasible with less running compressors. As we have seen, determining infeasibility seems to be easy and hence optimality can be proven shortly after. A partial explanation for the slower running times of minimizing the power loss

[1] Note that with the objective to minimize the power loss, i.e., min $\sum q_u^{\pm} p_u$, two scenarios could not be solved due to "unresolved numerical troubles" in the LP-relaxation. Hence, there are only 4225 scenarios listed in Table 6.6 for this objective function.

is that the objective functions favors unphysical solutions. For example, consider a single pipe $a = (u, v)$ with positive flow q_a. Then the LP-relaxation produces solutions with p_u as small as possible and p_v as large as possible. However, since the pressure decreases in the direction of the flow, $p_u > p_v$ has to hold. Therefore, the LP-relaxation has to describe the gas physics more accurately until a feasible solution is found.

To maximize the sum of the pressure variables at the nodes does not have an interpretation as clear as the other objective functions. However, it provides a good trade-off between fast running times and still being able to observe a difference between finding a feasible and an optimal solution. On the one hand, when solving problem (4.18) as feasibility problem or with the objective to minimize the number of running compressor stations, we cannot observe the influence of our techniques on the time to optimality. On the other hand, minimizing the power loss takes much more time.

Remark 6.3. As an attempt to find a bug in our implementation, which led to infeasibility of almost all of the 4227 scenarios for network `GasLib-582`, we implemented the option to relax the lower or upper pressure bounds by a given amount. Note that by relaxing the lower and upper bound by one bar each, about 500 previously infeasible scenarios turn to be feasible; see Table 6.10. Then we tried to identify where the error was in the following two ways. First we tried to minimize the maximal distance to the original pressure bounds such that there is a feasible solution, that is, we added a slack variable α, the constraints $p_v - \overline{p}_v \leq \alpha$ and $\underline{p}_v - p_v \leq \alpha$ for all nodes $v \in \mathcal{V}$, and minimized α. Secondly, we added slack variables α_v for all nodes $v \in \mathcal{V}$ and then minimized the sum of the slack variables.

Even though we could identify the bug in this particular way, the two objectives can be used to find bottlenecks in the network. However, with either of these objectives problem (4.18) seems to be hard to solve with almost half of the scenarios running into the time limit without a feasible solution.

6.3.3 Influence of the Compressor Model

Gas cannot be transported over long distances without increasing the pressure level in compressor stations along the way. Thus, compressor stations and their respective model play an important role for the feasibility of a scenario. Therefore, we compare computational results for the two variants introduced in Section 4.2.6. That is, the simple *box constraint model* given by the constraints (4.15) and the box constraint model with the additional facets as proposed by Walther et al. [154] to derive a better approximation of the feasible operating range of a compressor station.

Table 6.8. Results for all `GasLib-582` scenarios using the idealized model (idealCS), the box constraint model (box) and the box constraint model with the additional facets (box+facets) for compressor stations.

cs-model	opt	feas	limit	inf	inf-presol
idealCS	2049	95	2	2081	2073
box	1784	349	0	2094	2080
box+facets	918	156	19	3134	2300

Table 6.9. Geometric mean solving times in seconds and total run time in hours for the results presented in Table 6.8.

cs-model	to opt	to first	to inf	total	total time [h]
idealCS	368.27	76.47	1.01	21.29	416.99
box	720.42	73.75	1.02	31.91	829.31
box+facets	600.03	104.32	2.56	11.31	417.32

Furthermore, we implemented the following idealized model to get an impression if either model provides an accurate approximation of the feasible operating ranges of compressor stations. Our idealized model for a compressor station $cs = (u, v)$ consists of the constraints

$$
\begin{aligned}
&0 \le q_{cs} \le \overline{q}_{cs}\, z_{cs},\\
&\underline{p}_{cs}\, z_{cs} + \underline{p}_u(1 - z_{cs}) \le p_u,\\
&p_v \le \overline{p}_{cs}\, z_{cs} + \overline{p}_v(1 - z_{cs}),\\
&\underline{\Delta}\, z_{cs} + (\underline{p}_v - \overline{p}_u)(1 - z_{cs}) \le p_v - p_u,\\
&p_v - p_u \le \overline{\Delta}\, z_{cs} + (\overline{p}_v - \underline{p}_u)(1 - z_{cs}).
\end{aligned}
\tag{6.2}
$$

Here, we have the variables p_u, p_v, q_{cs} and z_{cs} as usual and we use the variable bounds $\overline{q}_{cs}$, $\underline{p}_u$, $\overline{p}_u$, $\underline{p}_v$, $\overline{p}_v$, and the technical limits $\underline{p}_{cs}$, $\overline{p}_{cs}$ as specified in the network data of the instance `GasLib-582`. Furthermore, we have lower and upper bounds on the possible pressure increase $\underline{\Delta}$ and $\overline{\Delta}$, which can be chosen by the user (of our implementation). With this model, there can only be flow on the compressor station if the compressor station is running ($z_{cs} = 1$), otherwise the flow is not coupled with the pressure. The technical limits and the bounds on the pressure increase have to be satisfied only if the compressor station is active. Otherwise, the pressure variables are decoupled. Moreover, for this model we do not consider any resistors as part of the compressor station, but there still can be a bypass valve to allow flow in the reverse direction (from v to u).

The computational results and solving times for all scenarios for the network `GasLib-582` with those three compressor station models are summarized in Tables 6.8 and 6.9. For the idealized model we used the parameters $\underline{\Delta} = 0\,\mathrm{bar}$ and $\overline{\Delta} = 45\,\mathrm{bar}$ for each compressor station such that the idealized model provides a relaxation of the other models. The table shows that the models only have a minor influence on the number of scenarios which are infeasible after presolving. However on the one hand, the additional facets for the box constraint model have a huge impact on the number of feasible scenarios. With the additional facets we have over 1000 infeasible scenarios more than without. And on the other hand, the box constraint model without the additional facets already seems to be quite idealized, since the numbers of infeasible scenarios are almost the same for the idealized model and the box constraint model. Therefore, we use the box constrained model with the additional facets for the following test.

6.3.4 Comprehensive Performance Tests

In this section, we finally study the effect of using the combinatorial model $\mathcal{P}^{\mathrm{AS}\pm}$ and the flow tightening techniques from Section 6.2 on the performance of Algorithm 4.2. We proceed similar to Section 5.4.2 and present results for the different model variants in combination with and without the optimization-based bound tightening technique (OBBT) and the flow tightening based on bound propagation (BP). Therefore, recall the following model variants used to enhance problem (4.18):

- **NFD** *no* binary variables to represent flow directions;
- **FDO** with binary variables, but no flow conservation or dicycle inequalities;
- **CB** dicycle inequalities (5.12) for a cycle basis;
- **AC** dicycle inequalities (5.12) for all cycles;
- **FLC** binary flow conservation (5.9) and (5.10);
- **FLC+CB** variant FLC plus dicycle inequalities for a cycle basis;
- **FLC+AC** variant FLC plus dicycle inequalities for all cycles.

In Section 5.4.2, we presented results for these variants with the algebraic model for the gas flow for all 4227 scenarios. However, since we want to compare these variants with the different combinations of the flow tightening techniques and the computations with the ODE model take more time, we use a smaller test set of 200 scenarios for network `GasLib-582` here. We chose this test set as follows. First of all, we performed computations using CPLEX and SOPLEX for model variant FLC+AC and both flow tightening techniques for all scenarios; the results and solving times using CPLEX are shown in Tables 6.10 and 6.11. For these computations we relaxed the lower and upper pressure bounds at the nodes (which are specified in the network

Table 6.10. Results for network `GasLib-582` with relaxation of the lower and upper pressure bounds by one bar categorized in temperature classes. The results are produced with model variant FLC+AC and flow tightening by OBBT and BP.

temperature class	opt	feas	limit	inf	inf-presol	total
warm	296	34	1	52	0	383
mild	296	55	4	292	7	647
cool	298	101	16	785	157	1200
cold	305	39	8	760	315	1112
freezing	134	18	9	724	386	885
total	1329	247	38	2613	865	4227

Table 6.11. Geometric mean solving times in seconds and total run time in hours for the results presented in Table 6.10.

temperature class	to opt	to first	to inf	total	total time [h]
warm	588.01	94.28	39.15	480.33	104.83
mild	689.54	97.04	35.63	210.51	143.65
cool	746.98	99.43	18.61	77.79	215.92
cold	513.38	93.24	8.09	32.68	117.74
freezing	459.35	89.01	4.88	11.89	59.01
total	607.79	95.41	11.00	57.39	641.15

data) by one bar for two reasons. On the one hand, relaxing the pressure bounds makes identifying infeasible scenarios more difficult such that we can also test the effect of our methods on the solving process of infeasible scenarios. On the other hand, by relaxing the pressure bounds there are more feasible or optimal scenarios to choose from. Especially in the temperature class freezing there are 152 scenarios with a feasible solution instead of only one scenario. We remark that the geometric mean times to the first feasible solution and to optimality in Table 6.11 are very similar compared to the corresponding geometric mean times without relaxing the pressure variables in Table 6.2. The geometric mean times with relaxation of the pressure bounds are increased a little, but show the same characteristics, i.e., for temperature class cool the time to optimality is the largest and identifying infeasibility is easier for colder temperatures. Only the geometric mean times for temperature class freezing differ significantly. However, this is due to the fact that without relaxing the pressure bounds there is only one feasible scenario in this class. After performing these computations with CPLEX and SOPLEX we removed the scenarios with contradicting results; see also Section 6.3.1. In total there are 28 scenarios where one solver

Table 6.12. Comparison of the arithmetic mean flow bounds of pipes after presolving, for which the flow is not already fixed.

	OBBT & BP		BP		OBBT		neither	
variant	LB	UB	LB	UB	LB	UB	LB	UB
NFD	−196.73	232.05	−217.81	371.02	−413.36	364.50	−483.79	587.80
FDO	−101.01	154.56	−165.36	307.81	−279.16	265.10	−444.35	564.22
CB	−89.45	157.29	−160.27	303.23	−220.38	224.74	−399.92	521.61
AC	−73.43	181.02	−143.14	264.68	−200.45	304.09	−398.53	518.81
FLC	−98.09	149.57	−163.53	297.66	−269.57	248.29	−443.64	563.36
FLC+CB	−72.64	110.35	−129.89	278.76	−188.87	160.08	−372.10	484.71
FLC+AC	−13.94	85.73	−92.67	163.28	−9.24	83.38	−359.27	448.39

produced a feasible or optimal solution and the other returned infeasible and 33 scenarios where primal and dual bounds of the two LP-solvers differ by more than 0.1%. Finally, from each of the temperature classes warm, mild, cool, cold and freezing we randomly chose 10 scenarios which were solved to optimality and 10 scenarios which were proven to be infeasible by both LP-solvers. Thereby, we chose only scenarios where CPLEX and SOPLEX produce the same result such that we can be (relatively) sure that this result is correct and to avoid scenarios which are numerically problematic.

The results for all model variants and combinations of flow tightening techniques OBBT and BP can be seen in Tables 6.12 to 6.16. The Tables 6.12 and 6.13 show statistics on the flow variables of pipelines after presolving. Table 6.12 displays the arithmetic mean lower and upper flow bounds for pipelines whose flow is not already fixed and Table 6.13 displays the arithmetic mean number of pipelines with fixed flow

Table 6.13. Comparison of the mean number of fixed flows and fixed directions of pipes after presolving.

	OBBT & BP		BP		OBBT		neither	
variant	#dir	#flows	#dir	#flows	#dir	#flows	#dir	#flows
NFD	44.13	157.00	51.78	142.00	43.80	157.00	44.40	142.00
FDO	52.20	156.72	60.21	142.13	51.88	156.35	58.55	142.13
CB	59.39	155.56	66.52	142.17	58.62	155.72	65.94	142.17
AC	63.49	150.46	67.00	142.17	62.76	150.01	65.94	142.17
FLC	52.44	157.10	60.37	142.13	51.52	157.10	58.43	142.13
FLC+CB	61.80	155.17	70.45	142.17	60.68	155.17	68.72	142.17
FLC+AC	67.05	153.31	74.81	142.33	62.59	155.04	69.72	142.17

Table 6.14. Comparison of arithmetic mean number of branch-and-bound nodes for scenarios solved to optimality or infeasibility. The numbers are rounded up.

	OBBT & BP		BP		OBBT		neither	
variant	opt	inf	opt	inf	opt	inf	opt	inf
NFD	427631	197390	460243	86272	–	–	–	–
FDO	667040	3134	–	15764	630724	9672	16245	23897
CB	683321	2859	410455	16597	522858	5836	–	42851
AC	–	7508	–	5287	590781	11844	–	23989
FLC	269324	820	298947	637	324837	958	292147	3587
FLC+CB	277270	447	270101	936	247487	758	298967	7501
FLC+AC	180169	191	186762	481	209586	4673	239784	1794

direction "#dir" and the mean number of pipelines with fixed flow direction "#flows". Table 6.14 shows the arithmetic mean numbers of nodes in the branch-and-bound tree for the scenarios solved to optimality or proven to be infeasible. Then Table 6.15 shows the results for the 200 scenarios, i.e., the number of feasible scenarios solved to optimality "opt", the number of scenarios that ran into the time limit with a feasible solution "feas" or without a feasible solution "limit" and the number of scenarios proven to be infeasible "inf". Table 6.16 shows the corresponding geometric mean times to optimality "to opt", the geometric mean time until the first feasible solution was found "to feas", and the geometric mean time until infeasibility was proven "to inf". In this table the geometric mean times are rounded up to the next full second.

Altogether, we can say that using the combinatorial model $\mathcal{P}^{\text{AS}\pm}$ and the flow tightening techniques have a very positive effect on the performance of our implementation. With model FLC+AC the flow bounds after presolving are tighter than with model NFD, i.e., the model without any binary direction variables, independently of using the flow tightening techniques or not. In Section 5.4.2, we have seen that we could speed-up the total running time for the algebraic gas flow model by a factor of ~ 7.2 with model variant FLC+AC compared to model NFD. Here we can see that for the ODE model we can speed-up the total running times by a factor of ~ 4.5 when using neither OBBT nor BP, by a factor of ~ 5.9 when using OBBT, by a factor of ~ 5.7 when using BP, and by a factor of ~ 6.3 when using both OBBT and BP. But more importantly, with model FLC+AC, OBBT and BP we can solve all 200 scenarios to optimality or prove infeasibility, whereas with model NFD without OBBT and BP we can solve none of the scenarios. Moreover, the geometric mean times for model FLC+AC with OBBT and BP show that our algorithm works quite fast with a geometric mean time to optimality of nearly 9 minutes and a geometric mean time of only 13 seconds to infeasibility.

The (probably) main reason for the performance improvement is analogous to Section 5.4.2. Using the binary variables to represent the flow directions and the corresponding constraints yields better relaxations of the original problem. This leads to a more successful presolving, that is, tighter variable bounds. Additionally using the problem specific flow tightening methods OBBT and BP improves the variable bounds even more. This implies that the LP-relaxations in the branch-and-bound tree are stronger and the search space is smaller, which leads to smaller branch-and-bound trees. Hence, the algorithm is faster. That the sizes of the branch-and-bound trees actually behave like this can be seen in Table 6.14, which shows the arithmetic mean numbers of nodes in the trees for the scenarios solved to optimality or infeasibility. For example, this effect can be seen by comparing the arithmetic mean number of processed nodes of the branch-and-bound tree for variants NFD and FLC+AC with OBBT and BP. For variant NFD the arithmetic mean number of nodes it takes to prove optimality or infeasibility is 427631 and 197390 nodes, respectively. In contrast for variant FLC+AC the mean number of nodes are 180169 and only 191, respectively.

When studying the results more closely, we observe that flow tightening based on bound propagation has only a minor effect on the number of fixed flow directions and a negligible effect on the number of fixed flows; see Table 6.13. However, BP significantly tightens the lower and upper flow bounds after presolving and also the numbers of scenarios which are optimally solved or proven to be infeasible and the corresponding geometric mean times to optimality and infeasibility are improved by using BP – at least for the models containing the binary flow conservation constraints (5.9) and (5.10), i.e., the model variants FLC, FLC+CB and FLC+AC.

In Section 5.4.2, we observed that variant AC defines tighter LP-relaxations for the algebraic model than variant FLC, but the running times and number of solved scenarios with variant FLC are superior to those of variant AC. Here, Tables 6.15 and 6.16 also show that variant FLC performs much better than variant AC. Furthermore, the flow bounds in Table 6.12 still show that variant AC defines tighter LP-relaxations when *not* using OBBT. But when using OBBT then variant FLC defines tighter LP-relaxations. The reason for that is that variant AC seems to define relaxations which are hard to solve if the binary flow conservation constraints are not used as well. For variant AC, we can observe that OBBT takes much more time than with other variants and, in fact, for more than half of the scenarios OBBT ran into its time limit of 10 minutes. When using BP as well, then OBBT still ran into its time limit for almost half of the scenarios. This observation also explains the larger geometric times to the first feasible solution with variant AC in Table 6.16.

Table 6.15. Number of scenarios that were solved to optimality, terminated with a feasible solution, terminated without feasible solution, and were proven to be infeasible for all model variants and all combinations of flow tightening techniques.

	OBBT & BP				BP				OBBT				neither			
variant	opt	feas	limit	inf	opt	feas	limit	inf	opt	feas	limit	inf	opt	feas	limit	inf
NFD	4	16	118	62	2	1	154	43	0	0	200	0	0	0	200	0
FDO	1	85	15	99	0	90	11	99	3	92	6	99	1	94	7	98
CB	2	92	6	100	2	91	8	99	1	94	5	100	0	95	9	96
AC	0	95	5	100	0	99	3	98	3	93	4	100	0	97	7	96
FLC	49	49	2	100	54	44	2	100	49	48	3	100	41	57	2	100
FLC+CB	75	23	2	100	68	29	3	100	65	30	5	100	58	40	3	99
FLC+AC	100	0	0	100	94	6	0	100	93	6	2	99	87	13	4	96

Table 6.16. Geometric mean solving times until the scenarios were solved to optimality, the first feasible solution was found, or the scenario was proven to be infeasible. Solving times are in seconds, rounded up.

	OBBT & BP			BP			OBBT			neither		
variant	to opt	to first	to inf	to opt	to first	to inf	to opt	to first	to inf	to opt	to first	to inf
NFD	2235	1894	165	2274	1761	40	–	–	–	–	–	–
FDO	2656	237	120	–	136	29	2883	294	154	124	145	77
CB	2935	299	125	1839	128	26	2477	368	180	–	180	83
AC	–	556	205	–	158	22	2199	635	377	–	178	68
FLC	738	125	50	988	72	8	983	128	67	1018	63	19
FLC+CB	718	96	42	788	75	8	768	112	55	918	81	14
FLC+AC	541	89	13	553	53	4	624	97	50	749	50	11

Yet, to find an explanation why variant AC defines relaxations which are hard to solve requires future investigation.

We have seen that OBBT actually worsens the performance of our algorithm for model variant AC and it increases the geometric mean times to infeasibility as well due to its additional computational overhead. Nevertheless, OBBT significantly tightens the variable bounds and increases the number of fixed flows; see Tables 6.12 and 6.13. Note that the reason why OBBT fixes more flow directions for variant NFD than for variant FLC+AC is the following. Since we only perform OBBT on variables whose lower and upper bounds are more than $10.0\,\mathrm{kg\,s^{-1}}$ apart and the flow bounds with model FLC+AC are already tighter before OBBT is applied, we perform OBBT on fewer variables for model FLC+AC which leads to less variable fixings. Furthermore, OBBT improves the geometric mean running times and the number of solved scenarios for model variants FLC, FLC+CB and FLC+AC; see Tables 6.15 and 6.16.

Overall the results show that with model variant FLC+AC, OBBT and BP we can solve optimization problems on the network `GasLib-582` quite fast with a geometric mean time of about 9 minutes to prove optimality and less than 100 seconds to prove infeasibility. We point out that this network is a slightly distorted version of a real-world gas transport network. Moreover, this real-world network actually covers roughly one-fourth of Germany; see Schmidt et al. [126]. The same network has been used to present other global optimization approaches on the example of stationary gas transport, too. However, among other modelling differences they use an algebraic potential-based flow model instead of the more complex ODE model we use; for example, see Pfetsch et al. [111], Koch et al. [82], or Burlacu et al. [20].

CHAPTER 7

Conclusion and Outlook

In this thesis, we have developed a new spatial branch-and-bound algorithm for the global optimization of a particular class of mixed-integer nonlinear optimization problems which contain ODE constraints. The distinguishing feature of this class is the assumption that the optimization problem only depends on the boundary values of the ODEs and not on the solution at some intermediate point. This assumption and the structure of the class of optimization problems is motivated by the application of stationary gas transport where it suffices to know the pressure at the ends of the pipelines and the ODEs describe the gas flow along the pipelines. Moreover, this assumption sets this particular class of optimization problems and the spatial branch-and-bound algorithm which is based on this assumption apart from other mixed-integer optimal control problems and dynamical systems which are considered in the literature, for example, the optimal gear shifting of a car or controlling the process of chemical reactions.

To define relaxations of the ODE solutions for the use in spatial branch-and-bound we have first derived sufficient conditions such that numerical one-step methods for solving scalar parameter-dependent initial value problems define lower or upper bounds on the exact solution in Chapter 2. We have used the corresponding result, that is, Lemma 2.3, to specify conditions on the differential equation such that the explicit midpoint method, the second-order Taylor method and the trapezoidal rule produce lower or upper bounds on the ODE solution. Moreover, we have investigated conditions such that the input-output functions defined by these methods are convex or concave. Afterwards, we have seen that these three methods can be used to define convex lower and upper bounds on the solution of the stationary isothermal Euler equation with and without height differences, which is the particular ODE in our application on stationary gas transport.

Based on the assumption that such lower and upper bounding methods exist for a general ODE (system) we have developed an adaptive spatial branch-and-bound algorithm in Chapter 3. The main ideas of this algorithm are as follows. If the ODEs are uniquely solvable and we have an algebraic formula for the exact relation between parameters and boundary values, then we could replace the ODEs by that formula. Otherwise, if we do not have such a formula, then the lower and upper bounding methods define a tube around the analytical solution, i.e., a relaxation. Assuming that we can construct convex under- and concave overestimators of these methods for the use in an black-box solver in finite time we can use these methods in the spatial branch-and-bound method as follows. In a node of the branch-and-bound tree we first solve a convex relaxation of the original problem and then use the lower and upper bounding methods to check if we are close to an exact solution (δ-feasible). For that, we refine the discretization if necessary. If the solution is not δ-feasible we improve the under- or overestimator to cut off the solution of the relaxation, or if this is not possible we perform branching such that the solution can be cut off afterwards. We have shown that the resulting spatial branch-and-bound algorithm, i.e., Algorithm 3.3, terminates finitely under mild conditions on the under- and overestimators; see Theorem 3.8. Moreover, we discussed how to define a consistent δ-feasibility notion and how to extend this approach to adaptively change the discretization which is not possible in other *first-discretize-then-optimize* approaches.

In Chapter 4, we applied our general framework to stationary gas transport. For this application, we use the explicit midpoint method and the trapezoidal rule to compute lower and upper bounds on the gas flow on pipelines without height differences. We have seen that we can use these to methods to construct linear relaxations of the gas flow for an LP-based spatial branch-and-bound algorithm and proven that the necessary conditions are satisfied such that the algorithm terminates finitely. Afterwards, we discussed possible extensions to this algorithm and presented first numerical results for our implementation with the branch-and-bound framework SCIP.

Motivated by the observations that our initial relaxations of the gas flow were weak and that the flow on potential-based networks such as stationary gas networks is necessarily acyclic, we derived and investigated combinatorial models to describe acyclic flows in Chapter 5. We have studied the properties of these models and in particular the model $\mathcal{P}^{\mathrm{AS}\pm}$. This model exploits the knowledge on sources and sinks, and that the flow is acyclic. Moreover, we have shown that linear optimization over $\mathcal{P}^{\mathrm{AS}\pm}$ is NP-hard. Nevertheless, we can use this combinatorial model to speed-up the optimization over potential-based flows. Using the Weymouth equation to describe the gas flow, we can speed-up the optimization on the gas network instance `GasLib-582` by a factor of about 7 through using $\mathcal{P}^{\mathrm{AS}\pm}$.

Finally in Chapter 6, we discussed our implementation for the application on stationary gas transport with the branch-and-bound framework SCIP. We have introduced problem specific bound tightening methods and presented further numerical results. These results have also shown that the combinatorial models for acyclic flows have a big impact on the performance of our algorithm for the ODE constrained model. Furthermore, with the additional bound tightening methods we can solve optimization problems on the network `GasLib-582` quite fast with a geometric mean time of 9 minutes to optimality which we otherwise could not solve. Again, we point out that this network is based on a real-world network which covers nearly a quarter of Germany and thus is of relevant size.

Outlook

This thesis still leaves plenty of open questions and starting points for future research and investigations. In Chapter 2, we have mainly considered scalar ODEs and only presented a preliminary result on the extension of the results to ODE systems. Furthermore, we only discussed ideas how to construct lower and upper bounds on the solutions of ODEs if the necessary requirements for the explicit midpoint method, second-order Taylor method or trapezoidal rule are not satisfied. These ideas and the extension to ODE systems require future investigation. Moreover, it is ongoing work by Prof. Stefan Ulbrich and Kristina Janzen to construct lower and upper bounds on the isothermal Euler equations in the instationary case, that is, for PDEs.

In the relaxation Algorithms 3.2 and 4.1 we choose the "most infeasible" ODE constraints to resolve infeasibility of optimal solutions for the convex relaxation. However, in mixed-integer linear programming it is known that *most fractional branching* in most cases does not perform any better than randomly choosing the branching variable; see Achterberg et al. [2]. Thus, it can be worth to study other selection criteria and test their influence on the computations. For the example of stationary gas transport we can investigate if there are pipelines which are more important than others. For example, this can be pipelines which are bridges in the underlying graph or pipelines which close cycles in the network.

In Chapter 4, we only consider a stationary setting. Thus, it can be part of future research to extend our spatial branch-and-bound algorithm to solve instationary problems. A starting point for this is the article by Burlacu et al. [19]. The authors discretize an instationary problem such that the problem turns into a stationary problem for each time step. With minor changes to the derivation of their discretization, we can derive a model such that each time step is a stationary problem similar to problem (4.18).

Concerning the combinatorial models for acyclic flows there are several interesting open questions, too. In Proposition 5.28 we investigated the dimension of $\mathcal{P}^{\mathrm{AS}\pm}$ in the single-source and single-sink case, however, in the general case it is an open question if the conditions in Lemma 5.18 suffice to define the dimension of $\mathcal{P}^{\mathrm{AS}\pm}$. Moreover, it would be interesting to obtain additional facets of $\mathcal{P}^{\mathrm{AS}\pm}$, for example, by transferring facets defined by k-fences and Möbius-ladders from the acyclic subgraph polytope if possible. Besides, from a computational point of view it would be interesting the compare the performance of our algorithm where we add the dicycle inequalities up front to an algorithm which dynamically separates dicycle inequalities. This can for example be done by considering a graph corresponding to the flow direction variables which are part of an LP-solution. If we can find a dicycle in this graph, then we can cut off the LP-solution via the corresponding dicycle inequality. This is of particular interest for solving even larger gas networks where enumerating all cycles might not be possible any more.

In the last chapter, we compared numerical results for different models for compressor stations and have seen that the box constraint model (4.15) is rather idealized. Thus, we could improve our code by implementing a more detailed or realistic model for compressor stations. Moreover, we observed in Section 6.3.4 that adding all dicycle constraints (5.12) but not the binary flow conservation constraints (5.9) and (5.10), renders even the simple relaxation without nonlinear constraints, which we use for OBBT, hard to solve. It requires future investigation why that is the case. Solving this issue might even lead to improvements for the model variants FLC+AC using both the dicycle inequalities and the flow conservation constraints.

Bibliography

[1] T. Achterberg, *Conflict analysis in mixed integer programming*, Discrete Optimization, 4 (2007), pp. 4–20. doi: `10.1016/j.disopt.2006.10.006`. [→152]

[2] T. Achterberg, T. Koch, and A. Martin, *Branching rules revisited*, Operations Research Letters, 33 (2005), pp. 42–54. doi: `10.1016/j.orl.2004.04.002`. [→62, 101, 179]

[3] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas, *A global optimization method, αBB, for general twice differentiable NLPs – II. Implementation and computational results*, Computers & Chemical Engineering, 22 (1998), pp. 1159–1179. doi: `10.1016/S0098-1354(98)00218-X`. [→11, 52, 56, 64, 89]

[4] C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier, *A global optimization method, αBB, for general twice differentiable NLPs – I. Theoretical advances*, Computers & Chemical Engineering, 22 (1998), pp. 1137–1158. doi: `10.1016/S0098-1354(98)00027-1`. [→11, 52, 56, 64, 89]

[5] B. Assarf, E. Gawrilow, K. Herr, M. Joswig, B. Lorenz, A. Paffenholz, and T. Rehn, *Computing convex hulls and counting integer points with* **polymake**, Mathematical Programming Computputation, 9 (2017), pp. 1–38. doi: `10.1007/s12532-016-0104-z`. [→128]

[6] I. Bajaj and M. M. F. Hasan, *Global dynamic optimization using edge-concave underestimator*, Journal on Global Optimization, 77 (2020), pp. 487–512. doi: `10.1007/s10898-020-00883-2`. [→12, 53]

[7] K.-H. Becker and B. Hiller, *ASTS orientations on undirected graphs: Structural analysis and enumeration*, ZIB-Report 18-31, Zuse Institute Berlin, 2018. [→13, 113, 114, 128, 132]

[8] K.-H. Becker and B. Hiller, *Improved optimization models for potential-driven network flow problems via ASTS orientations*, ZIB-Report 19-58, Zuse Institute Berlin, 2019. [→13, 113, 114, 128, 132, 142, 160]

[9] K.-H. Becker and B. Hiller, *Efficient enumeration of acyclic graph orientations with sources or sinks revisited*, ZIB-Report 20-05, Zuse Institute Berlin, 2020. [→13, 113, 114]

[10] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan, *Mixed-integer nonlinear optimization*, Acta Numerica, 22 (2013), pp. 1–131. doi: 10.1017/S0962492913000032. [→51]

[11] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter, *Branching and Bounds Tightening Techniques for Non-Convex MINLP*, Optimization Methods and Software, 24 (2009), pp. 597–634. doi: 10.1080/10556780903087124. [→56, 60, 89, 155]

[12] D. Bienstock, *Electrical Transmission System Cascades and Vulnerability: An Operations Research Viewpoint*, MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015. doi: 10.1137/1.9781611974164. [→116]

[13] G. Birkhoff and J. B. Diaz, *Non-linear network problems*, Quarterly of Applied Mathematics, 13 (1956), pp. 431–443. doi: 10.1090/qam/77398. [→113]

[14] *Bundesministerium für Bildung und Forschung*. https://www.bmbf.de/. Last visited: June 25, 2020. [→2]

[15] H. G. Bock, C. Kirches, A. Meyer, and A. Potschka, *Numerical solution of optimal control problems with explicit and implicit switches*, Optimization Methods and Software, 33 (2018), pp. 450–474. doi: 10.1080/10556788.2018.1449843. [→52]

[16] P. Bonami, J. Lee, S. Leyffer, and A. Wächter, *More branch-and-bound experiments in convex nonlinear integer programming*, tech. report, Optimization Online, 2011, http://www.optimization-online.org/DB_HTML/2011/09/3191.html. [→101]

[17] C. Buchheim, R. Kuhlmann, and C. Meyer, *Combinatorial optimal control of semilinear elliptic PDEs*, Computational Optimization and Applications, 70 (2018), pp. 641–675. doi: 10.1007/s10589-018-9993-2. [→12]

[18] J. Burgschweiger, B. Gnädig, and M. C. Steinbach, *Optimization models for operative planning in drinking water networks*, Optimization and Engineering, 10 (2009), pp. 43–73. doi: 10.1007/s11081-008-9040-8. [→116]

[19] R. Burlacu, H. Egger, M. Gross, A. Martin, M. E. Pfetsch, L. Schewe, M. Sirvent, and M. Skutella, *Maximizing the storage capacity of gas networks: a global MINLP approach*, Optimization and Engineering, 20 (2019), pp. 543–573. doi: 10.1007/s11081-018-9414-5. [→179]

[20] R. Burlacu, B. Geissler, and L. Schewe, *Solving mixed-integer nonlinear programmes using adaptively refined mixed-integer linear programmes*,

Optimization Methods and Software, 35 (2020), pp. 37–64. doi: 10.1080/10556788.2018.1556661. [→14, 160, 176]

[21] B. Chachuat, A. B. Singer, and P. I. Barton, *Global mixed-integer dynamic optimization*, AIChE Journal, 51 (2005), pp. 2235–2253. doi: 10.1002/aic.10494. [→12, 16, 53, 58]

[22] B. Chachuat, A. B. Singer, and P. I. Barton, *Global Methods for Dynamic Optimization and Mixed-Integer Dynamic Optimization*, Industrial & Engineering Chemistry Research, 45 (2006), pp. 8373–8392. doi: 10.1021/ie0601605. [→12, 16, 53, 58]

[23] B. Chachuat and M. Villanueva, *Bounding the Solutions of Parametric ODEs: When Taylor Models Meet Differential Inequalities*, in 22nd European Symposium on Computer Aided Process Engineering, I. D. L. Bogle and M. Fairweather, eds., vol. 30 of Computer Aided Chemical Engineering, Elsevier, 2012, pp. 1307–1311. doi: 10.1016/B978-0-444-59520-1.50120-2. [→17]

[24] M. Čižniar, M. Podmajerský, T. Hirmajer, M. Fikar, and A. M. Latifi, *Global optimization for parameter estimation of differential-algebraic systems*, Chemical Papers, 63 (2009), pp. 274–283. doi: 10.2478/s11696-009-0017-7. [→52]

[25] C. F. Colebrook, *Turbulent Flow in Pipes, with particular reference to the Transition Region between the Smooth and Rough Pipe Laws*, Journal of the Institution of Civil Engineers, 11 (1939), pp. 133–156. doi: 10.1680/ijoti.1939.13150. [→104]

[26] M. Collins, L. Cooper, R. Helgason, J. Kennington, and L. LeBlanc, *Solving the pipe network analysis problem using optimization techniques*, Management Science, 24 (1978), pp. 747–760. doi: 10.1287/mnsc.24.7.747. [→13, 115]

[27] Y. Crama, *Concave extensions for nonlinear 0–1 maximization problems*, Mathematical Programming, 61 (1993), pp. 53–60. doi: 10.1007/BF01582138. [→92]

[28] H. Diedam and S. Sager, *Global optimal control with the direct multiple shooting method*, Optimal Control Applications and Methods, 39 (2018), pp. 449–470. doi: 10.1002/oca.2324. [→52]

[29] P. Domschke, B. Geissler, O. Kolb, J. Lang, A. Martin, and A. Morsi, *Combination of Nonlinear and Linear Optimization of Transient Gas Networks*, INFORMS Journal on Computing, 23 (2011), pp. 605–617. doi: 10.1287/ijoc.1100.0429. [→76]

[30] M. A. Duran and I. E. Grossmann, *An outer-approximation algorithm for a class of mixed-integer nonlinear programs*, Mathematical Programming, 36 (1986), pp. 307–339. doi: 10.1007/BF02592064. [→31, 52, 56, 66, 89, 91]

[31] W. R. Esposito and C. A. Floudas, *Deterministic Global Optimization in Nonlinear Optimal Control Problems*, Journal of Global Optimization, 17 (2000), pp. 97–126. doi: 10.1023/A:1026578104213. [→53]

[32] J. Fakcharoenphol, B. Laekhanukit, and P. Sukprasert, *Finding all useless arcs in directed planar graphs*, 2018, https://arxiv.org/abs/1702.04786. [→133]

[33] J. E. Falk, *Lagrange Multipliers and Nonconvex Programs*, SIAM Journal on Control, 7 (1969), pp. 534–545. doi: 10.1137/0307039. [→51, 92]

[34] J. E. Falk and K. R. Hoffman, *A Successive Underestimation Method for Concave Minimization Problems*, Mathematics of Operations Research, 1 (1976), pp. 251–259. doi: 10.1287/moor.1.3.251. [→51, 93]

[35] J. E. Falk and R. M. Soland, *An Algorithm for Separable Nonconvex Programming Problems*, Management Science, 15 (1969), pp. 550–569. doi: 10.1287/mnsc.15.9.550. [→51, 92]

[36] M. Feistauer, *Mathematical methods in fluid dynamics*, vol. 67 of Pitman Monographs and Surveys in Pure and Applied Mathematics, Longman Scientific & Technical, 1993. [→3]

[37] E. J. Finnemore and J. B. Franzini, *Fluid Mechanics with Engineering Applications*, vol. 10, McGraw-Hill New York, 2002. [→104]

[38] C. A. Floudas and C. E. Gounaris, *A review of recent advances in global optimization*, Journal of Global Optimization, 45 (2008), pp. 3–38. doi: 10.1007/s10898-008-9332-8. [→51]

[39] S. Fortune, J. Hopcroft, and J. Wyllie, *The directed subgraph homeomorphism problem*, Theoretical Computer Science, 10 (1980), pp. 111–121. doi: 10.1016/0304-3975(80)90009-2. [→133]

[40] G. Gamrath, D. Anderson, K. Bestuzheva, W.-K. Chen, L. Eifler, M. Gasse, P. Gemander, A. Gleixner, L. Gottwald, K. Halbig, G. Hendel, C. Hojny, T. Koch, P. L. Bodic, S. J. Maher, F. Matter, M. Miltenberger, E. Mühmer, B. Müller, M. E. Pfetsch, F. Schlösser, F. Serrano, Y. Shinano, C. Tawfik, S. Vigerske, F. Wegscheider, D. Weninger, and J. Witzig, *The SCIP Optimization Suite 7.0*, tech. report, Optimization Online, 2020, http://www.optimization-online.org/DB_HTML/2020/03/7705.html. [→56, 89, 101, 107, 141, 147, 159]

[41] GasLib, *GasLib – a library of gas network instances.* http://gaslib.zib.de/. [→107, 141, 147, 160]

[42] *Gasunie*. https://www.gasunie.nl/en. Last visited: June 25, 2020. [→2]

[43] E. Gawrilow and M. Joswig, *polymake: a framework for analyzing convex polytopes*, in Polytopes—combinatorics and computation (Oberwolfach, 1997), vol. 29 of DMV Seminar, Birkhäuser, Basel, 2000, pp. 43–73. doi: 10.1007/978-3-0348-8438-9_2. [→128]

[44] B. Geissler, A. Morsi, L. Schewe, and M. Schmidt, *Solving power-constrained gas transportation problems using an MIP-based alternating direction method*, Computers & Chemical Engineering, 82 (2015), pp. 303–317. doi: 10.1016/j.compchemeng.2015.07.005. [→76]

[45] B. Geissler, A. Martin, A. Morsi, and L. Schewe, *The MILP-relaxation approach*, in Evaluating Gas Network Capacities, T. Koch, B. Hiller, M. E. Pfetsch, and L. Schewe, eds., MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015, Chapter 6, pp. 103–122. doi: 10.1137/1.9781611973693.ch6. [→5, 76, 81]

[46] M. Gerdts, *Solving mixed-integer optimal control problems by branch-and-bound: a case study from automobile test-driving with gear shift*, Optimal Control Applications and Methods, 26 (2005), pp. 1–18. doi: 10.1002/oca.751. [→11]

[47] A. Gleixner, T. Berthold, B. Müller, and S. Weltge, *Three enhancements for optimization-based bound tightening*, Journal of Global Optimization, 67 (2017), pp. 731–757. doi: 10.1007/s10898-016-0450-4. [→142, 155]

[48] S. Göttlich, A. Potschka, and U. Ziegler, *Partial Outer Convexification for Traffic Light Optimization in Road Networks*, SIAM Journal on Scientific Computing, 39 (2017), pp. B53–B75. doi: 10.1137/15M1048197. [→12]

[49] M. Gross, M. E. Pfetsch, L. Schewe, M. Schmidt, and M. Skutella, *Algorithmic results for potential-based flows: Easy and hard cases*, Networks, 73 (2019), pp. 306–324. doi: 10.1002/net.21865. [→115, 116]

[50] M. Grötschel, M. Jünger, and G. Reinelt, *On the acyclic subgraph polytope*, Mathematical Programming, 33 (1985), pp. 28–42. doi: 10.1007/BF01582009. [→124, 125]

[51] M. Gugat, F. Hante, M. Hirsch-Dick, and G. Leugering, *Stationary States in Gas Networks*, Networks and Heterogeneous Media, 10 (2015), pp. 295–320. doi: 10.3934/nhm.2015.10.295. [→77, 89]

[52] M. Gugat, G. Leugering, A. Martin, M. Schmidt, M. Sirvent, and D. Wintergerst, *MIP-based instantaneous control of mixed-integer PDE-constrained gas transport problems*, Computational Optimization and Applications, 70 (2018), pp. 267–294. doi: 10.1007/s10589-017-9970-1. [→76]

[53] M. Gugat, G. Leugering, A. Martin, M. Schmidt, M. Sirvent, and D. Wintergerst, *Towards simulation based mixed-integer optimization with differential equations*, Networks, 72 (2018), pp. 60–83. doi: 10.1002/net.21812. [→12, 53, 76, 160]

[54] M. Gugat, R. Schultz, and D. Wintergerst, *Networks of pipelines for gas with nonconstant compressibility factor: stationary states*, Computational and Applied Mathematics, 37 (2018), pp. 1066–1097. doi: 10.1007/s40314-016-0383-z. [→8, 15, 55, 77, 89]

[55] O. Habeck and M. E. Pfetsch, *Combinatorial Acyclicity Models for Potential-based Flows*, tech. report, Optimization Online, 2020, http://www.optimization-online.org/DB_HTML/2020/04/7761.html. [→13, 113]

[56] O. Habeck, M. E. Pfetsch, and S. Ulbrich, *Global Optimization of Mixed-Integer ODE Constrained Network Problems Using the Example of Stationary Gas Transport*, SIAM Journal on Optimization, 29 (2019), pp. 2949–2985. doi: 10.1137/17M1152668. [→13, 16, 47, 51, 75, 114]

[57] M. Hahn, C. Kirches, P. Manns, S. Sager, and C. Zeile, *Decomposition and approximation for PDE-constrained mixed-integer optimal control*, in SPP1962 Special Issue, M. H. et al., ed., Birkhäuser, 2019, https://mathopt.de/PUBLICATIONS/Hahn2019.pdf. [→12]

[58] F. M. Hante, G. Leugering, A. Martin, L. Schewe, and M. Schmidt, *Challenges in Optimal Control Problems for Gas and Fluid Flow in Networks of Pipes and Canals: From Modeling to Industrial Applications*, in Industrial Mathematics and Complex Systems: Emerging Mathematical Models, Methods and Algorithms, P. Manchanda, R. Lozi, and A. H. Siddiqi, eds., Springer, Singapore, 2017, pp. 77–122. doi: 10.1007/978-981-10-3758-0_5. [→2, 12, 75]

[59] F. M. Hante and S. Sager, *Relaxation methods for mixed-integer optimal control of partial differential equations*, Computational Optimization and Applications, 55 (2013), pp. 197–225. doi: 10.1007/s10589-012-9518-3. [→12]

[60] F. M. Hante and M. Schmidt, *Convergence of Finite-Dimensional Approximations for Mixed-Integer Optimization with Differential Equations*, tech. report, Optimization Online, 2019, http://www.optimization-online.org/DB_HTML/2018/12/6973.html. [→6, 11, 53]

[61] P. Hartman, *Ordinary Differential Equations*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2002. doi: 10.1137/1.9780898719222. [→54]

[62] S. M. Harwood and P. I. Barton, *Efficient polyhedral enclosures for the reachable set of nonlinear control systems*, Mathematics of Control, Signals,

and Systems, 28 (2016), pp. 1–33. doi: 10.1007/s00498-015-0153-2. [→17, 57]

[63] S. M. Harwood and P. I. Barton, *Affine relaxations for the solutions of constrained parametric ordinary differential equations*, Optimal Control Applications and Methods, 39 (2018), pp. 427–448. doi: 10.1002/oca.2323. [→12, 17, 57]

[64] M. M. F. Hasan, *An edge-concave underestimator for the global optimization of twice-differentiable nonconvex problems*, Journal of Global Optimization, 71 (2018), pp. 735–752. doi: 10.1007/s10898-018-0646-x. [→12, 53]

[65] C. Hayn, H. Heitsch, R. Henrion, H. Leövey, A. Möller, and W. Römisch, *Methods for verifying booked capacities*, in Evaluating Gas Network Capacities, T. Koch, B. Hiller, M. E. Pfetsch, and L. Schewe, eds., MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015, Chapter 14, pp. 291–315. doi: 10.1137/1.9781611973693.ch14. [→161]

[66] R. Hemmecke, M. Köppe, J. Lee, and R. Weismantel, *Nonlinear Integer Programming*, in 50 Years of Integer Programming 1958–2008: From the Early Years to the State-of-the-Art, M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, eds., Springer, Berlin, Heidelberg, 2010, pp. 561–618. doi: 10.1007/978-3-540-68279-0_15. [→51]

[67] C. T. Hendrickson and B. N. Janson, *A common network flow formulation for several civil engineering problems*, Civil Engineering Systems, 1 (1984), pp. 195–203. doi: 10.1080/02630258408970343. [→12, 113, 115]

[68] B. Hiller and K. H. Becker, *Improving relaxations for potential-driven network flow problems via acyclic flow orientations*, ZIB-Report 18-30, Zuse Institute Berlin, 2018. [→13, 113, 116, 128]

[69] B. Hiller and T. Walther, *Modelling compressor stations in gas networks*, Tech. Report 17-67, Zuse Institut Berlin, 2017. [→79, 85, 159]

[70] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE constraints*, vol. 23 of Mathematical Modelling: Theory and Applications, Springer, New York, 2009. doi: 10.1007/978-1-4020-8839-1. [→12, 52]

[71] P. Hofer, *Beurteilung von Fehlern in Rohrnetzberechnungen (Error evaluation in calculation of pipelines)*, GWF–Gas/Erdgas, 11 (1973), pp. 113–119. [→104]

[72] R. Horst, *An algorithm for nonconvex programming problems*, Mathematical Programming, 10 (1976), pp. 312–321. doi: 10.1007/BF01580678. [→51]

[73] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, Springer, Berlin, Heidelberg, 3 ed., 1996. doi: 10.1007/978-3-662-03199-5. [→11, 32, 40, 48, 51, 56, 57]

[74] B. Houska, M. E. Villanueva, and B. Chachuat, *A validated integration algorithm for nonlinear ODEs using Taylor models and ellipsoidal calculus*, in 52nd IEEE Conference on Decision and Control, 2013, pp. 484–489. doi: 10.1109/CDC.2013.6759928. [→17]

[75] B. Houska, M. E. Villanueva, and B. Chachuat, *Stable Set-Valued Integration of Nonlinear Dynamic Systems using Affine Set-Parameterizations*, SIAM Journal on Numerical Analysis, 53 (2015), pp. 2307–2328. doi: 10.1137/140976807. [→17]

[76] M. N. Jung, G. Reinelt, and S. Sager, *The Lagrangian relaxation for the combinatorial integral approximation problem*, Optimization Methods and Software, 30 (2015), pp. 54–80. doi: 10.1080/10556788.2014.890196. [→12, 52]

[77] B. Kalantari and J. B. Rosen, *An Algorithm for Global Minimization of Linearly Constrained Concave Quadratic Functions*, Mathematics of Operations Research, 12 (1987), pp. 544–561. doi: 10.1287/moor.12.3.544. [→92]

[78] R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, R. Miller and J. Thatcher, eds., Plenum Press, 1972, pp. 85–103. [→125]

[79] M. R. Kılınç and N. V. Sahinidis, *Chapter 21: State of the art in mixed-integer nonlinear optimization*, in Advances and Trends in Optimization with Engineering Applications, T. Terlaky, M. F. Anjos, and S. Ahmed, eds., MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017, pp. 273–292. doi: 10.1137/1.9781611974683.ch21. [→11, 51, 56]

[80] C. Kirches, F. Lenders, and P. Manns, *Approximation Properties and Tight Bounds for Constrained Mixed-Integer Optimal Control*, SIAM Journal on Control and Optimization, 58 (2020), pp. 1371–1402. doi: 10.1137/18M1182917. [→12, 52]

[81] K. Kleibohm, *Bemerkungen zum Problem der nichtkonvexen Programmierung*, Unternehmensforschung, 11 (1967), pp. 49–60. doi: 10.1007/BF01922383. [→51, 92]

[82] T. Koch, B. Hiller, M. E. Pfetsch, and L. Schewe, eds., *Evaluating Gas Network Capacities*, MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015. doi: 10.1137/1.9781611973693. [→4, 6, 12, 14, 76, 79, 85, 103, 115, 160, 176]

[83] B. Korte and J. Vygen, *Combinatorial Optimization. Theory and Algorithms*, vol. 21 of Algorithms and Combinatorics, Springer, Berlin, Heidelberg, 6th ed., 2018. [→121, 133]

[84] J. Králik, P. Stiegler, Z. Vostrý, and J. Záworka, *Dynamic Modeling of Large-Scale Networks with Application to Gas Distribution*, vol. 6 of Studies in Automation and Control, Elsevier, 1988. [→4]

[85] J. Lang and J. G. Verwer, *On global error estimation and control for initial value problems*, SIAM Journal on Scientific Computing, 29 (2007), pp. 1460–1475. doi: `10.1137/050646950`. [→19]

[86] B. E. Larock, R. W. Jeppson, and G. Z. Watters, *Hydraulics of pipeline systems*, CRC press, 2010. [→116]

[87] J. Lee and S. Leyffer, eds., *Mixed Integer Nonlinear Programming*, vol. 154 of IMA Volumes in Mathematics and its Applications, Springer, New York, 2012. doi: `10.1007/978-1-4614-1927-3`. [→51]

[88] K. Lehmann, A. Grastien, and P. Van Hentenryck, *The Complexity of DC-Switching Problems*, 2014, `https://arxiv.org/abs/1411.4369`. [→119]

[89] L. Liberti and C. C. Pantelides, *Convex envelopes of monomials of odd degree*, Journal of Global Optimization, 25 (2003), pp. 157–168. doi: `10.1023/A:1021924706467`. [→52, 56, 89]

[90] Y. Lin, J. A. Enszer, and M. A. Stadtherr, *Enclosing all solutions of two-point boundary value problems for ODEs*, Computers & Chemical Engineering, 32 (2008), pp. 1714–1725. doi: `10.1016/j.compchemeng.2007.08.013`. [→17]

[91] Y. Lin and M. A. Stadtherr, *Deterministic global optimization of nonlinear dynamic systems*, AIChE Journal, 53 (2007), pp. 866–875. doi: `10.1002/aic.11101`. [→12, 16, 53]

[92] Y. Lin and M. A. Stadtherr, *Validated solutions of initial value problems for parametric ODEs*, Applied Numerical Mathematics, 57 (2007), pp. 1145–1162. doi: `10.1016/j.apnum.2006.10.006`. [→17]

[93] M. Locatelli and F. Schoen, *Global Optimization*, MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2013. doi: `10.1137/1.9781611972672`. [→51, 56, 58, 62]

[94] D. Mahlke, A. Martin, and S. Moritz, *A mixed integer approach for time-dependent gas network optimization*, Optimization Methods and Software, 25 (2010), pp. 625–644. doi: `10.1080/10556780903270886`. [→76]

[95] P. Manns and C. Kirches, *Improved regularity assumptions for partial outer convexification of mixed-integer PDE-constrained optimization problems*, ESAIM: COCV, 26 (2020), p. 32. doi: `10.1051/cocv/2019016`. [→12]

[96] A. Martin, M. Möller, and S. Moritz, *Mixed Integer Models for the Stationary Case of Gas Network Optimization*, Mathematical Programming, 105 (2006), pp. 563–582. doi: 10.1007/s10107-005-0665-5. [→76]

[97] R. Mattheij and J. Molenaar, *Ordinary Differential Equations in Theory and Practice*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2002. doi: 10.1137/1.9780898719178. [→18, 22, 39]

[98] J. J. Maugis, *Etude de réseaux de transport et de distribution de fluide*, RAIRO Operations Research, 11 (1977), pp. 243–248. [→13, 115]

[99] G. P. McCormick, *Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems*, Mathematical Programming, 10 (1976), pp. 147–175. doi: 10.1007/BF01580665. [→11, 52, 56, 64, 89]

[100] K. Mehlhorn, A. Neumann, and J. M. Schmidt, *Certifying 3-edge-connectivity*, Algorithmica, 77 (2017), pp. 309–335. doi: 10.1007/s00453-015-0075-x. [→133]

[101] C. A. Meyer and C. A. Floudas, *Convex envelopes for edge-concave functions*, Mathematical Programming, 103 (2005), pp. 207–224. doi: 10.1007/s10107-005-0580-9. [→52, 56, 89, 93]

[102] R. Misener and C. A. Floudas, *ANTIGONE: Algorithms for coNTinuous/Integer Global Optimization of Nonlinear Equations*, Journal of Global Optimization, 59 (2014), pp. 503–526. doi: 10.1007/s10898-014-0166-2. [→56, 89]

[103] R. E. Moore, *Methods and Applications of Interval Analysis*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979. doi: 10.1137/1.9781611970906. [→17]

[104] N. Nedialkov, K. Jackson, and G. Corliss, *Validated solutions of initial value problems for ordinary differential equations*, Applied Mathematics and Computation, 105 (1999), pp. 21–68. doi: 10.1016/S0096-3003(98)10083-8. [→11, 17]

[105] M. Neher, K. R. Jackson, and N. S. Nedialkov, *On Taylor model based integration of ODEs*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 236–262. doi: 10.1137/050638448. [→11, 17]

[106] J. Nikuradse, *Strömungsgesetze in rauhen Rohren*, Forschungsheft auf dem Gebiete des Ingenieurwesens, VDI-Verlag, Düsseldorf, 1933. [→4, 103]

[107] J. Nikuradse, *Laws of Flow in Rough Pipes*, vol. Technical Memorandum 1292, National Advisory Committee for Aeronautics Washington, 1950. [→4, 103]

[108] A. J. Osiadacz, *Different transient models – limitations, advantages and disadvantages*, tech. report, PSIG report 9606, Pipeline Simulation Interest Group, 1996. [→5]

[109] I. Papamichail and C. S. Adjiman, *A rigorous global optimization algorithm for problems with ordinary differential equations*, Journal of Global Optimization, 24 (2002), pp. 1–33. doi: 10.1023/A:1016259507911. [→53]

[110] I. Papamichail and C. S. Adjiman, *Proof of convergence for a global optimization algorithm for problems with ordinary differential equations*, Journal of Global Optimization, 33 (2005), pp. 83–107. doi: 10.1007/s10898-004-6100-2. [→53]

[111] M. E. Pfetsch, A. Fügenschuh, B. Geissler, N. Geissler, R. Gollmer, B. Hiller, J. Humpola, T. Koch, T. Lehmann, A. Martin, A. Morsi, J. Rövekamp, L. Schewe, M. Schmidt, R. Schultz, R. Schwarz, J. Schweiger, C. Stangl, M. C. Steinbach, S. Vigerske, and B. M. Willert, *Validation of nominations in gas network optimization: models, methods, and solutions*, Optimization Methods and Software, 30 (2015), pp. 15–53. doi: 10.1080/10556788.2014.888426. [→14, 160, 176]

[112] Y. Puranik and N. V. Sahinidis, *Domain reduction techniques for global NLP and MINLP optimization*, Constraints, 22 (2017), pp. 338–376. doi: 10.1007/s10601-016-9267-5. [→60, 155]

[113] A. D. Rikun, *A convex envelope formula for multilinear functions*, Journal of Global Optimization, 10 (1997), pp. 425–437. doi: 10.1023/A:1008217604285. [→52, 93]

[114] R. Z. Ríos-Mercado and C. Borraz-Sánchez, *Optimization problems in natural gas transportation systems: A state-of-the-art review*, Applied Energy, 147 (2015), pp. 536–555. doi: 10.1016/j.apenergy.2015.03.017. [→5, 12, 76]

[115] R. Z. Ríos-Mercado, S. Wu, L. R. Scott, and E. A. Boyd, *A Reduction Technique for Natural Gas Transmission Network Optimization Problems*, Annals fo Operations Research, 117 (2002), pp. 217–234. doi: 10.1023/A:1021529709006. [→13, 115]

[116] R. T. Rockafellar, *Convex analysis*, vol. 28 of Princeton Mathematical Series, Princeton Univeristy Press, Princeton, 1970. doi: 10.1515/9781400873173. [→92, 93]

[117] R. T. Rockafellar, *Network Flows and Monotropic Optimization*, Athena Scientific, Belmont, Massachusetts, 2nd ed., 1998. [→113, 115]

[118] H. S. Ryoo and N. V. Sahinidis, *A branch-and-reduce approach to global optimization*, Journal of Global Optimization, 8 (1996), pp. 107–138. doi: 10.1007/BF00138689. [→155]

[119] S. Sager, H. G. Bock, and M. Diehl, *The integer approximation error in mixed-integer optimal control*, Mathematical Programming, 133 (2012), pp. 1–23. doi: 10.1007/s10107-010-0405-3. [→12, 52]

[120] S. Sager, H. G. Bock, and G. Reinelt, *Direct methods with maximal lower bound for mixed-integer optimal control problems*, Mathematical Programming, 118 (2009), pp. 109–149. doi: 10.1007/s10107-007-0185-6. [→12, 52]

[121] S. Sager, M. Jung, and C. Kirches, *Combinatorial integral approximation*, Mathematical Methods of Operationl Research, 73 (2011), pp. 363–380. doi: 10.1007/s00186-011-0355-4. [→12, 52]

[122] S. Sager and C. Zeile, *On Mixed-Integer Optimal Control with Constrained Total Variation of the Integer Control*, tech. report, Optimization Online, 2019, http://www.optimization-online.org/DB_HTML/2019/10/7432.html. [→12, 52]

[123] N. V. Sahinidis, *BARON: A general purpose global optimization software package*, Journal of Global Optimization, 8 (1996), pp. 201–205. doi: 10.1007/BF00138693. [→56, 89, 155]

[124] A. Sahlodin and B. Chachuat, *Convex/concave relaxations of parametric ODEs using Taylor models*, Computers & Chemical Engineering, 35 (2011), pp. 844–857. doi: 10.1016/j.compchemeng.2011.01.031. [→11, 17]

[125] A. M. Sahlodin and B. Chachuat, *Discretize-then-relax approach for convex/concave relaxations of the solutions of parametric ODEs*, Applied Numerical Mathematics, 61 (2011), pp. 803–820. doi: 10.1016/j.apnum.2011.01.009. [→11, 17]

[126] M. Schmidt, D. Assmann, R. Burlacu, J. Humpola, I. Joormann, N. Kanelakis, T. Koch, D. Oucherif, M. E. Pfetsch, L. Schewe, R. Schwarz, and M. Sirvent, *GasLib – A Library of Gas Network Instances*, Data, 2 (2017). doi: 10.3390/data2040040. [→107, 141, 147, 160, 176]

[127] M. Schmidt, M. Sirvent, and W. Wollner, *The Cost of Not Knowing Enough: Mixed-Integer Optimization with Implicit Lipschitz Nonlinearities*, tech. report, Optimization Online, 2018, http://www.optimization-online.org/DB_HTML/2018/04/6594.html. [→77]

[128] M. Schmidt, M. Sirvent, and W. Wollner, *A decomposition method for MINLPs with Lipschitz continuous nonlinearities*, Mathematical Programming, 178 (2019), pp. 449–483. doi: 10.1007/s10107-018-1309-x. [→12, 53, 76, 160]

[129] M. Schmidt, M. C. Steinbach, and B. M. Willert, *High detail stationary optimization models for gas networks*, Optimization and Engineering, 16 (2015), pp. 131–164. doi: 10.1007/s11081-014-9246-x. [→77, 85]

[130] A. Schrijver, *Finding k disjoint paths in a directed planar graph*, SIAM Journal on Computing, 23 (1994), pp. 780–788. doi: 10.1137/S0097539792224061. [→133]

[131] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, vol. 24 of Algorithms and Combinatorics, Springer, Berlin, Heidelberg, 2003. [→134]

[132] SCIP, *Solving Constraint Integer Programs*. http://scip.zib.de/. [→101, 107, 141, 147, 159]

[133] J. K. Scott and P. I. Barton, *Convex and Concave Relaxations for the Parametric Solutions of Semi-explicit Index-One Differential-Algebraic Equations*, Journal of Optimization Theory and Applications, 156 (2013), pp. 617–649. doi: 10.1007/s10957-012-0149-8. [→17, 53]

[134] J. K. Scott and P. I. Barton, *Improved relaxations for the parametric solutions of ODEs using differential inequalities*, Journal of Global Optimization, 57 (2013), pp. 143–176. doi: 10.1007/s10898-012-9909-0. [→12, 17, 57]

[135] J. K. Scott, B. Chachuat, and P. I. Barton, *Nonlinear convex and concave relaxations for the solutions of parametric ODEs*, Optimal Control Applications and Methods, 34 (2013), pp. 145–163. doi: 10.1002/oca.2014. [→53, 57]

[136] J. K. Scott, M. D. Stuber, and P. I. Barton, *Generalized McCormick relaxations*, Journal of Global Optimization, 51 (2011), pp. 569–606. doi: 10.1007/s10898-011-9664-7. [→17, 52, 57]

[137] H. D. Sherali and A. Alameddine, *A new reformulation-linearization technique for bilinear programming problems*, Journal of Global optimization, 2 (1992), pp. 379–410. doi: 10.1007/BF00122429. [→11, 52, 56, 89]

[138] H. D. Sherali and C. H. Tuncbilek, *A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique*, Journal of Global Optimization, 2 (1992), pp. 101–112. doi: 10.1007/BF00121304. [→11, 52, 56, 89]

[139] H. D. Sherali and C. H. Tuncbilek, *A reformulation-convexification approach for solving nonconvex quadratic programming problems*, Journal of Global Optimization, 7 (1995), pp. 1–31. doi: 10.1007/BF01100203. [→11, 52, 56, 89]

[140] A. B. Singer and P. I. Barton, *Bounding the solutions of parameter dependent nonlinear ordinary differential equations*, SIAM Journal on Scientific Computing, 27 (2006), pp. 2167–2182. doi: 10.1137/040604388. [→12, 17, 57]

[141] A. B. Singer and P. I. Barton, *Global optimization with nonlinear ordinary differential equations*, Journal of Global Optimimization, 34 (2006), pp. 159–190. doi: 10.1007/s10898-005-7074-4. [→53]

[142] M. Sirvent, *Incorporating Differential Equations into Mixed-Integer Programming for Gas Transport Optimization*, phd thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2018. doi: 10.25593/978-3-96147-114-0. [→77]

[143] R. D. Skeel, *Thirteen ways to estimate global error*, Numerische Mathematik, 48 (1986), pp. 1–20. doi: 10.1007/BF01389440. [→19]

[144] E. Smith and C. Pantelides, *A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs*, Computers & Chemical Engineering, 23 (1999), pp. 457–478. doi: 10.1016/S0098-1354(98)00286-5. [→15]

[145] M. D. Stuber, J. K. Scott, and P. I. Barton, *Convex and concave relaxations of implicit functions*, Optimization Methods and Software, 30 (2015), pp. 424–460. doi: 10.1080/10556788.2014.924514. [→52, 57]

[146] J. Szabó, *The set of solutions to nomination validation in passive gas transportation networks with a generalized flow formula*, ZIB-Report 11-44, Zuse Intitute Berlin, 2012. [→119]

[147] F. Tardella, *On the existence of polyhedral convex envelopes*, in Frontiers in Global Optimization, C. A. Floudas and P. Pardalos, eds., Boston, Massachusetts, 2004, Springer US, pp. 563–573. doi: 10.1007/978-1-4613-0251-3. [→52, 56, 89, 92, 93]

[148] R. E. Tarjan, *A note on finding the bridges of a graph*, Information Processing Letters, 2 (1974), pp. 160–161. doi: 10.1016/0020-0190(74)90003-9. [→133]

[149] M. Tawarmalani, J.-P. P. Richard, and C. Xiong, *Explicit convex and concave envelopes through polyhedral subdivisions*, Mathematical Programming, 138 (2013), pp. 531–577. doi: 10.1007/s10107-012-0581-4. [→52, 89, 93, 94]

[150] *Vereinigung der Fernleitungsnetzbetreiber Gas e.V.* https://www.fnb-gas.de/. Last visited: June 25, 2020. [→2, 77]

[151] S. Vigerske and A. Gleixner, *SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework*, Optimization Methods and Software, 33 (2018), pp. 563–593. doi: 10.1080/10556788.2017.1335312. [→148]

[152] M. E. Villanueva, B. Houska, and B. Chachuat, *Unified framework for the propagation of continuous-time enclosures for parametric nonlinear*

ODEs, Journal of Global Optimization, 62 (2015), pp. 575–613. doi: 10.1007/s10898-014-0235-6. [→17]

[153] W. Walter, *Differential and Integral Inequalities*, Springer, Berlin, Heidelberg, 1970. doi: 10.1007/978-3-642-86405-6. [→17]

[154] T. Walther, B. Hiller, and R. Saitenmacher, *Polyhedral 3d models for compressors in gas networks*, Tech. Report 17-66, Zuse Institut Berlin, 2017. [→85, 86, 159, 168]

[155] M. E. Wilhelm, A. V. Le, and M. D. Stuber, *Global optimization of stiff dynamical systems*, AIChE Journal, 65 (2019). doi: 10.1002/aic.16836. [→52]

[156] C. Zeile, N. Robuschi, and S. Sager, *Mixed-Integer Optimal Control under Minimum Dwell Time Constraints*, tech. report, Optimization Online, 2019, http://www.optimization-online.org/DB_HTML/2019/09/7366.html. [→12, 52]

[157] C. Zeile, T. Weber, and S. Sager, *Combinatorial Integral Approximation Decompositions for Mixed-Integer Optimal Control*, tech. report, Optimization Online, 2018, http://www.optimization-online.org/DB_HTML/2018/02/6472.html. [→12, 52]

List of Algorithms

List of Figures

List of Tables

Wissenschaftlicher Werdegang

06/2015 – 09/2020	Wissenschaftlicher Mitarbeiter am Fachbereich Mathematik der Technischen Universität Darmstadt in der Arbeitsgruppe Diskrete Optimierung und im Sonderforschungsbereich Transregio 154 „Mathematische Modellierung, Simulation und Optimierung am Beispiel von Gasnetzwerken“
04/2015	Abschluss Master of Science in Mathematik
10/2009 – 04/2015	Studium der Mathematik an der Technischen Universität Darmstadt
06/2008	Abitur am Friedrich Alexander Gymnasium in Neustadt an der Aisch