Mathematical Programming for Stable Control and Safe Operation of Gas Transport Networks

vorgelegt von
M. Sc.
Kai Hoppmann-Baum
ORCID: 0000-0001-9184-8215

an der Fakultät II – Mathematik und Naturwissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
Dr. rer. nat.

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Michael Scheutzow
Gutachter: Prof. Dr. Thorsten Koch
Gutachter: Prof. Dr. Martin Schmidt

Tag der wissenschaftlichen Aussprache: 30.05.2022

Berlin 2022

The chase is better than the catch.
-Hans Peter Geerdes

Abstract

The fight against climate change makes extreme but inevitable changes in the energy sector necessary. These in turn lead to novel and complex challenges for the transmission system operators (TSOs) of gas transport networks. In this thesis, we consider four different planning problems emerging from real-world operations and present mathematical programming models and solution approaches for all of them.

Due to regulatory requirements and side effects of renewable energy production, controlling today's gas networks with their involved topologies is becoming increasingly difficult. Based on the "network station" modeling concept for approximating the technical capabilities of complex subnetworks, e.g., compressor stations, we introduce a tri-level mixed-integer program to determine important global control decisions. Its goal is to avoid changes in the network elements' settings while deviations from future inflow pressures as well as supplies and demands are minimized. A sequential linear programming inspired post-processing routine is run to derive physically accurate solutions w.r.t. the transient gas flow in pipelines. Computational experiments based on real-world data show that meaningful solutions are quickly and reliably determined. Therefore, the algorithmic approach is used within KOMPASS, a decision support system for the transient network control that we developed with the Open Grid Europe GmbH (OGE), one of Europe's largest natural gas TSOs.

Anticipating future use cases, we adapt the aforementioned algorithmic approach for hydrogen transport. We investigate whether the natural gas infrastructure can be repurposed and how the network control changes when energy-equivalent amounts of hydrogen are transported. Besides proving the need for purpose-built compressors, we observe that, due to the reduced linepack, the network control becomes more dynamic, compression energy increases by 440% on average, and stricter regulatory rules regarding the balancing of supply and demand become necessary.

Extreme load flows expose the technical limits of gas networks and are therefore of great importance to the TSOs. In this context, we introduce the Maximum Transportation Problem and the Maximum Potential Transport Moment Problem to determine severe transport scenarios. Both can be modeled as linear bilevel programs where the leader selects supplies and demands, maximizing the follower's transport effort. To solve them, we identify solution-equivalent instances with acyclic networks, provide variable bounds regarding their KKT reformulations, apply the big-M technique, and solve the resulting MIPs. A case study shows that the obtained scenarios exceed the maximum severity values of a provided test set by at least 23%.

OGE's transmission system is 11,540 km long. Monitoring it is crucial for safe operations. To this end, we discuss the idea of using uncrewed aerial vehicles and introduce the Length-Constrained Cycle Partition Problem to optimize their routing. Its goal is to find a smallest cycle partition satisfying vertex-induced length requirements. Besides a greedy-style heuristic, we propose two MIP models. Combining them with symmetry-breaking constraints as well as valid inequalities and lower bounds from conflict hypergraphs yields a highly performant solution algorithm for this class of problems.

Zusammenfassung

Der Kampf gegen den Klimawandel erfordert eine umfassende Neuausrichtung des gesamten Energiesektors. Dies wiederum stellt die Betreiber von Gastransportnetzen (Transmission System Operators, TSOs) vor beträchtliche Herausforderungen. In dieser Arbeit befassen wir uns mit diversen Planungsproblemen aus der Praxis und den sich daraus ergebenden mathematischen Programmen: der Steuerung des Gasflusses, der Bewertung des Netzwerks hinsichtlich extremer Transportszenarien, sowie der Kontrolle ihrer Funktionstüchtigkeit mittels unbemannter Luftfahrzeuge (uncrewed aerial vehicles, UAVs). Darüber hinaus widmen wir uns dem immer wichtiger werdenden Transport von Wasserstoff, welcher zusätzliche Anforderungen an die bereits bestehende Infrastruktur stellt.

Die Steuerung heutiger Gastransportnetze mit ihren komplexen Topologien wird aufgrund schwieriger regulatorischer Rahmenbedingungen und Schwankungen in der erneuerbaren Energieerzeugung immer komplizierter. In diesem Zusammenhang stellen wir ein gemischt-ganzzahliges Trilevel-Programm zur Bestimmung wichtiger globaler Steuerentscheidungen vor. Dieses basiert auf dem Modellierungskonzept der "Netzstationen", welches der Approximation der technischen Steuerungsmöglichkeiten von komplexen Teilnetzen mit Verdichterstationen dient. Ziel ist es, das Umschalten von Netzelementen zu vermeiden, während gleichzeitig Abweichungen von den vorgegebenen Eingangsdrücken sowie den Ein- und Ausspeisungen minimiert werden. Zur Erzeugung physikalisch korrekter Lösungen bezüglich des transienten Gasflusses in Rohren haben wir eine Postprocessing-Routine entwickelt, die auf Sequentieller Linearer Programmierung beruht. Rechenexperimente mit realen Daten zeigen, dass mit diesem Verfahren praktisch einsetzbare Lösungen schnell und zuverlässig ermittelt werden. Unser algorithmischer Ansatz wird daher mittlerweile in KOMPASS eingesetzt, einem Decision-Support-System für die transiente Netzsteuerung, welches wir zusammen mit einem Entwicklerteam der Open Grid Europe GmbH (OGE), einem der größten europäischen Erdgas-Fernleitungsnetzbetreiber, implementiert haben.

Im Hinblick auf zukünftige Anwendungsgebiete untersuchen wir, wie sich der beschriebene algorithmische Ansatz auf die Steuerung von Wasserstofftransportnetzen übertragen lässt. Wir untersuchen, ob die Erdgasinfrastruktur umgewidmet werden kann und wie sich die Netzsteuerung verändert, wenn energieäquivalente Wasserstoffmengen transportiert werden. Zum einen weisen wir nach, dass für einen solchen Einsatz speziell angefertigte Verdichter notwendig sein werden. Zum anderen stellen wir fest, dass die Netzsteuerung aufgrund des reduzierten Netzpuffers dynamischer wird, die Gasverdichtung im Schnitt 440% mehr Energie benötigt und strengere Regulierungsmaßnahmen hinsichtlich der Bilanzierung von Ein- und Ausspeisung erforderlich sein werden.

Extreme Lastflüsse zeigen die technischen Grenzen der Gastransportnetze auf und sind daher von großer Bedeutung für die TSOs. Wir stellen das Maximum Transportation Problem und das Maximum Potential Transport Moment Problem vor, welche der Bestimmung schwieriger Transportszenarien dienen. Beide Probleme kön-

nen als lineare Bilevel-Programme modelliert werden, bei denen der Leader Ein- und Ausspeisungen wählt, welche den Transportaufwand des Followers maximieren. Um diese Optimierungsprobleme zu lösen, identifizieren wir sogenannte lösungsäquivalente Probleminstanzen mit azyklischen Netzwerken, liefern Schranken an die Variablen ihrer KKT-Reformulierungen, wenden die Big-M-Methode an und lösen die daraus resultierenden gemischt-ganzzahligen Programme. Eine Fallstudie zeigt, dass die ermittelten Szenarien die maximalen Schwierigkeitsgrade eines vorgegebenen Testsatzes um mindestens 23% übertreffen.

Das Fernleitungsnetz von OGE ist 11540 km lang und seine Überwachung ist von entscheidender Bedeutung für den sicheren Betrieb. Dies kann zum Beispiel durch den Einsatz von UAVs gewährleistet werden. In diesem Kontext untersuchen wir deren optimale Streckenführung und führen das Length-Constrained Cycle Partition Problem ein. Ziel ist es, eine kleinste Kreispartition zu finden, die knoteninduzierten Längenanforderungen genügt. Neben einer Greedy-Heuristik stellen wir zwei gemischt-ganzzahlige Optimierungsmodelle vor. Kombinieren wir diese mit symmetriebrechenden Nebenbedingungen sowie gültigen Ungleichungen und unteren Schranken, die von Konflikt-Hypergraphen abgeleitet werden, erhalten wir einen sehr effektiven Algorithmus für diese Klasse von Problemen.

Contents

1. Introduction

During my time as a PhD student, I was a passionate participant in science slams. These events are similar to poetry slams, but instead of reciting self-written poems or texts, the contestants give ten-minute talks about their research to an ordinary audience. In the end, the crowd gets to vote on which presentation it liked best.

Niemann et al. [117] have investigated the perception of the audience and the reasons for attending science slams. In essence, this combination of scientific content and entertainment, often referred to as "edutainment", is considered an excellent way to impart knowledge. Their results coincide in many aspects with my personal observations. After having attended many contests, I identified three characteristics of winning presentations, which seem to have a positive influence on the final vote: The level of entertainment and fun, the individual relatability of the listeners to the topic, and the subject's impact on society.

The design of an entertaining, relatable, and impactful ten-minute talk, "Mathematical Programming for Stable Control and Safe Operation of Gas Transport Networks", turned out to be quite challenging. Of course, a certain amount of fun can be brought in by insinuating lavatorial humor. However, I can report that this wins you some pitying smiles at best. On the other hand, it is hard to identify any individual relationship that Jane or John Doe may have to gas transport networks, except for heating and cooking maybe, not to mention their nonrelation to mathematical programming. Thus, all stakes were put on the third horse and I argued why gas transport networks are essential for our society: To ensure the security of energy supply and because of their crucial role in the fight against climate change. After explaining the underlying connections, the audiences became highly interested in how our mathematical-programming based algorithmic approaches can help to enable a stable control and safe operation of these networks, which are necessary prerequisites to ensure the former and succeed in the latter.

Besides winning some of the slams, the presentation motivated people in the audience to approach me after the contests, ask further questions, and engage in discussions on the matter. This interest enormously strengthened my motivation as it confirmed that this branch of research is indeed of great social relevance. On the other hand, it also proved that many people are unfortunately not aware of this.

In this thesis, we address various challenges that currently arise in the context of real-world gas transport networks. We introduce related problem formulations, optimization models based on mathematical programming, and algorithmic approaches to solve them efficiently. Simultaneously, we aim to raise awareness for the critical issues related to these challenges and the beautiful concept of mathematical programming, which we apply to improve upon the networks' control and operation.

Figure 1.1.: The author of this thesis presenting at a science slam in Halle (Saale) in October 2020. For more information regarding science slams, we refer to scienceslam.de [148]. Paedagogic-oriented discussions can be found in the work of Stimm [117] and Eisenbarth and Weißkopf [154].

Natural Gas Transport Networks – Importance and Challenges

With nearly 25% in the German energy mix (2017), natural gas constitutes one of the country's primary energy sources. Heating accounts for the largest share in the annual demand of approximately 95 billion cubic meters [44], which is about two times the volume of Bodensee lake [53]. Additionally, natural gas plays a vital role in the chemical industry and electricity generation as stated by the German Federal Ministry for Economic Affairs and Energy [44].

The answer to how natural gas is delivered to the demand sites was given decades ago when the first transport networks have been established. A pipeline between Bentheim and Chemische Werke Hüls, built in 1938, marked the beginning of the natural gas transport era [121]. Over the years, the German gas network developed into a European "hub" due to its geographical location. Today, a great majority of more than 90% of the natural gas consumed in Germany is imported from Russia, Norway, and the Netherlands, and nearly 50% of the natural gas entering Germany is exported to neighboring countries again, mainly to South-Western Europe [17].

Thus, providing a stable control and safe operation of natural gas networks is necessary to ensure the security of the energy supply, not only for Germany. However, due to several reasons, this task is becoming increasingly difficult.

Growing populations and economies led to drastically increasing energy and natural gas demands [115], which made extensions, changes, and updates of the infrastructure necessary. Many initially simple network structures evolved into complex and involved topologies, resulting in a substantial increase in technical control options. These control measures are, in addition, highly interdependent, which makes

the task of controlling these networks more challenging. This observation is further underlined by our explanations in Section 3.1, where we give a brief high-level introduction to the physics and control of gas transport networks.

Another reason adding to the increasing difficulty regarding the network control is that, even though experts expect that the overall natural gas consumption will remain constant or decrease in the near future [145], the hourly supplies and demands at the entries and exits are becoming more volatile, as reported by practitioners.

A possible explanation for this behavior is the market model that the European Union implemented in order to decouple natural gas trading and transport [42, 43]. Nowadays, the transport requests for the upcoming day are established twenty-four hours in advance through nominations of the gas traders. Thereby, they possess a wide range of options, which complicates long-term planning and makes reliable forecasting of future inflows and outflows harder. In this context, the question of what the possible worst-case transport scenarios look like arises.

The growing usage of green energy, e.g., solar and wind power, is a second reason for the fluctuations in supply and demand. Due to the planned nuclear and coal phase-outs, the share of renewables in the energy mix will continue to increase in the future. However, there is some natural uncertainty regarding their production. One way to mitigate this effect is to use natural gas-fired power plants, which can be ramped up on short notice [128]. Thus, the TSOs must be prepared for spontaneously occurring peak demands at the corresponding sites.

Finally, there are rather unpredictable events that impede a stable network control and its safe operation. Prime examples are political tensions leading to an interruption of supply [166] or malfunctioning hardware [160], which makes a standard control impossible. In this context, it is up to the TSOs to take preventive actions wherever possible, e.g., by monitoring the transport infrastructure on a regular basis in order to avoid severe failures.

Hydrogen Transport Networks – Backbone of Future Energy Supply

An indispensable part of any policy to tackle climate change is a drastic decrease in greenhouse gas emissions. Thereby, all commonly accepted strategies envisage a complete replacement of fossil energy carriers. Applying some variant of the so-called power-to-gas (P2G) concept, where (surplus) renewable electric energy is used to produce an intermediate fuel gas, is an essential ingredient. This fuel gas can be stored and reconverted in times of low electricity supply and high demand or directly be used in the industry. Depending on the gas and the production technique, the P2G process can be designed environmentally friendly [157].

Due to its versatility, the currently favored choice for the fuel gas is hydrogen, which is underlined by the national hydrogen strategy of the German government [33]. While hydrogen already serves as an alternative fuel in the automotive area, further potential exists in the mobility sector, e.g., the aviation [10] or shipping industry [134]. Moreover, it could for example also be used in the steel industry to replace fossil energy carriers [124].

Naturally, the problem of where to store and how to transport the hydrogen arises. Here, pipeline networks and connected storage facilities seem to be the most suitable approach for supplying densely populated areas with enormous demands, see for example the studies of Yang and Ogden [164] and Reuß et al. [137]. However, planning and building new pipeline infrastructures is an expensive and time-consuming task. Hence, the seemingly obvious-looking idea of repurposing the existing billion Euro valued natural gas transport infrastructure is gaining much attention [2, 35, 70]. The cost of such a conversion is expected to be about 10–15% of a construction from scratch, whose implementation process would additionally take up five to seven years from initial planning to commissioning in the best case [2]. Moreover, hydrogen transport represents a future business model for natural gas TSOs, and the European Hydrogen Backbone Initiative is evidence of this [159].

Current Relevant Questions from Practice

All the topics discussed above are at the heart of the GasLab and the EnergyLab of the Research Campus MODAL (Mathematical Optimization and Data Analysis Laboratories) [113], a public-private partnership project funded by the Federal Ministry of Education and Research [46]. In its context, we were approached by the experts and practitioners of Open Grid Europe GmbH (OGE) [120], one of Europe's largest natural gas TSOs. They had realized that novel operations research and mathematical optimization techniques are necessary to solve their complex and so far intractable planning problems. After intensive discussions, we pinned down four essential questions, which summarize the crucial real-world challenges that OGE and other natural gas TSOs currently face in daily operations.

- ▷ How can a TSO guarantee a stable network control?
- ▷ Can the natural gas network be repurposed for hydrogen transport?
- ▷ What are possible worst-case transport scenarios that can arise?
- ▷ How can the transport infrastructure be monitored efficiently?

1.1. Outline and Contribution

Our work regarding these four questions defines the structure of this thesis. After introducing basic notation and essential concepts from mathematical programming, graph theory, and flow networks in Chapter 2, we discuss each question in detail, formulate a corresponding optimization problem as a mathematical programming based model, and develop a solution approach in Chapters 3, 4, and 5. To this end, we focus on large-scale networks and real-world data provided by our industry partner since our results are required to be applicable in practice and shall be applied in everyday operations at OGE. In the following, we give a detailed overview of the content of the three chapters by summarizing our contribution w.r.t. the four questions. A short general conclusion in Chapter 6 marks the end of the thesis.

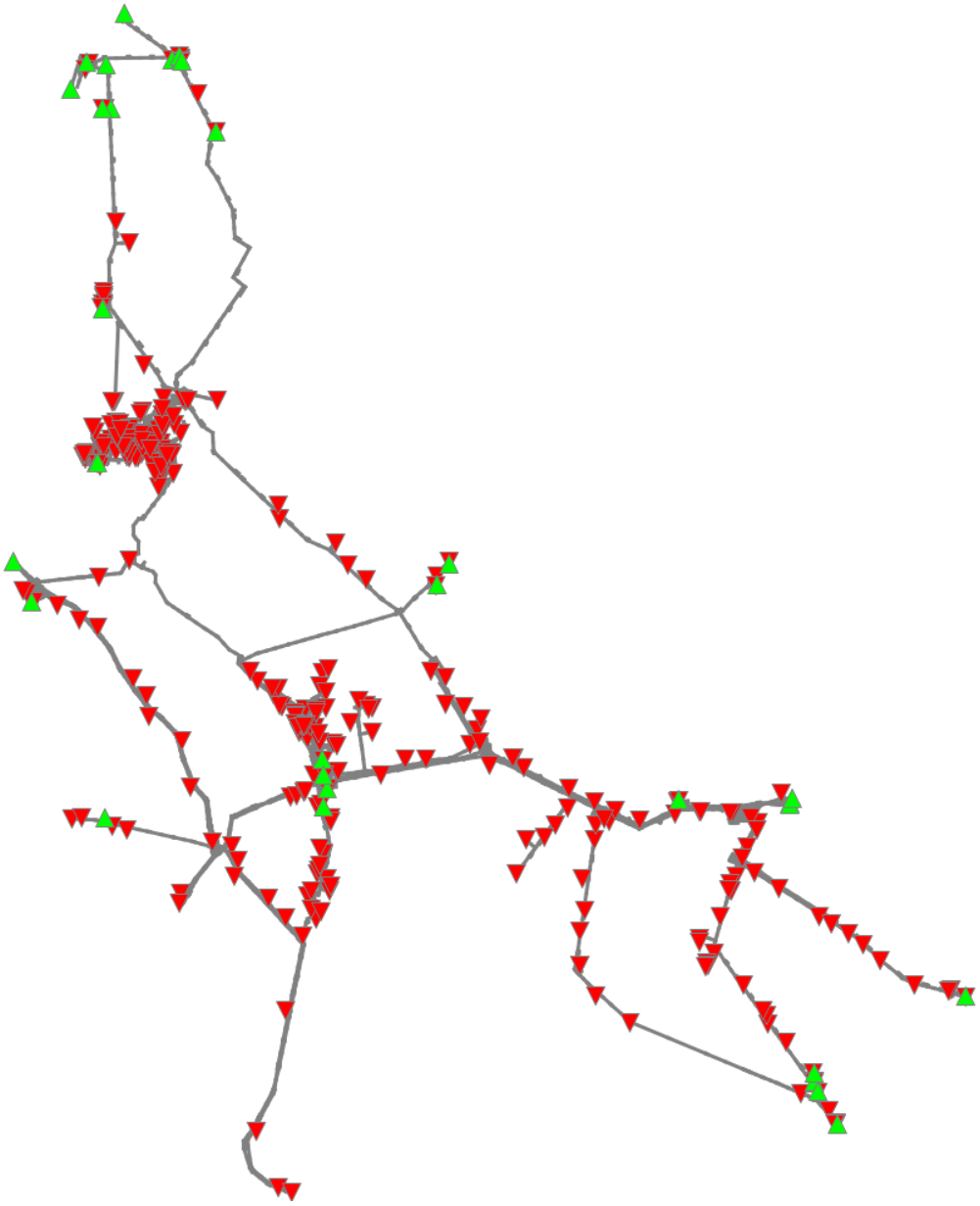


Figure 1.2.: Picture of OGE's natural gas transport network, which serves as the basis for our research in the GasLab and EnergyLab of research campus MODAL. While entry points are shown as green triangles ▲, exit points are depicted as red triangles ▼. Moreover, grey lines represent segments of pipelines.

How can a TSO guarantee a stable network control?

The control of gas transport networks is a challenging task incorporating many technical and regulatory constraints, but necessary to ensure the security of the energy supply. However, due to the increasingly complex network topologies, the design of the natural gas market, and interfering but inevitable side effects of renewable energy production, providing a stable control is becoming increasingly difficult.

In this context, we present KOMPASS (Kontinuierliches Optimierungsmodul zur prognoseabgesicherten Systemsteuerung / Continuous optimization module for a prognosis-based system control) in Chapter 3: The first decision support system for dispatchers controlling natural gas networks. Based on current forecasts of future flows and pressures, KOMPASS continuously recommends control measures to ensure a stable network control. In the following, we present its architecture and discuss those parts in depth to which the research presented in this thesis contributed.

Given the network topology, technical and physical data related to its entities, and the current state in terms of gas flows and pressures, future supplies and demands for the entry and exit points are predicted using the approach of Petkovic et al. [129]. Subsequently, additional time series for the pressure values at the entries are heuristically generated based on the predicted supplies.

Afterward, in the first stage of KOMPASS, which constitutes a central topic of Chapter 3, we propose an algorithmic approach for determining a stable transient control for gas transport networks. It consists of a tri-level MIP model, followed by a sequential linear programming inspired postprocessing routine.

The tri-level MIP is designed to make important global control decisions, e.g., how to route the flow and where and when to compress the gas. The objectives of the two upper levels are to ensure the existence of a feasible technical network control while deviations from the predicted pressures and flows are minimized. The goal of the third level is then to actually determine a feasible technical network control using as few technical measures as possible, i.e., changes in the operation modes of network elements. The latter is considered a meaningful indicator for stability w.r.t. the network control by practitioners. To solve the tri-level MIP, we introduce an algorithm based on solving a sequence of closely related single-level MIPs.

To obtain a computationally tractable formulation, we apply several approximations of the physics of gas transport networks. For example, we use the concept of network stations to model complex pipeline intersection areas, which comprise multiple compressor stations, regulators, and valves. Thereby, the technical capabilities of each network station, e.g., the increase of gas pressure through compression, are given by a purpose-built and hand-tailored directed graph representation. The interplay of its arcs is then described using the concepts of so-called flow directions and simple states. In particular, for each network station and each time step, we must choose a flow direction, which determines where the gas enters and leaves, and a supporting simple state, which enables the usage of a subset of the artificial arcs.

Moreover, we apply a linear model for the transient gas flow through pipelines. To eliminate possible drawbacks regarding its physical accuracy, we propose an iterative

velocity adjustment procedure (IVAP) as postprocessing routine, which is inspired by sequential linear programming.

Our algorithmic approach yields promising results in computational experiments, which we conducted using instances that are based on a major subnetwork of OGE's transport infrastructure and corresponding historically measured flow and pressure values. The experts consider the obtained solutions meaningful, and they are additionally quickly and reliably determined, which is a crucial property w.r.t. the underlying application, i.e., the usage within KOMPASS.

In the second and last stage of KOMPASS, the global control decisions recommended by the first stage algorithm are verified, and highly detailed technical control measures for the original network elements realizing them are determined. To this end, the approach proposed by Hennings et al. [78] is applied.

Can the natural gas network be repurposed for hydrogen transport?

Besides the previously discussed economic considerations regarding a transformation, existing studies have mainly focused on the feasibility w.r.t. the technical components, e.g., whether the installed pipelines are suited for hydrogen transport, see for example Peter et al. [2], Dodds and Demoullin [35], and Haeseldonckx and D'haeseleer [70]. However, although the crucial question of how the control of such a network changes compared to natural gas transport has been raised [35], it has not been investigated yet. This thesis constitutes, as far as we know, the first work in this direction and reveals several interesting insights.

In the course of Chapter 3, we explicitly state our algorithmic approach for determining a stable transient network control for hydrogen networks, too. To investigate whether the natural gas infrastructure can be repurposed and how its control changes when energy-equivalent amounts of hydrogen are transported, we discuss physical and technical properties that differ w.r.t. the two gases. Based on this, we propose a method for converting the natural gas input data into hydrogen transport scenarios. Using the converted natural gas instances, we demonstrate that replacing the currently installed turbo compressors with special hydrogen compressor units is necessary to allow for feasible technical network control. This is because the maximum compression ratios of the former are drastically reduced. Due to the reduced linepack, we also observe that more technical measures need to be conducted as the hydrogen must be stored in or retrieved from more remote parts of the network. For the same reasons, the amount of compression energy increases by 440% on average. Finally, it is necessary to enforce stricter regulatory rules regarding the balancing of supply and demand to transport energy-equivalent amounts of hydrogen.

What are possible worst-case transport scenarios that can arise?

When discussing stable network control, a natural question is what the most challenging problem instances look like. Indeed, the identification of severe transport scenarios is a crucial task in the context of determining the technical capacity of a

gas transport network, i.e., the maximum amounts of gas it can transport.

Chapter 4 starts with an explanation of the regulatory rules for the German natural gas market, which is a necessary prerequisite to understand how transport scenarios arise in the first place. Afterward, we review several methods from the literature as well as two approaches that are used in practice to determine a network's technical capacity, i.e., the amounts of gas it can transport. Both of the latter are based on evaluating a finite set of severe transport scenarios, which is designed to cover and represent a wide range of possible but difficult flow situations. Our reviews demonstrate that it is beneficial to consider multiple, diverse severity measures when selecting transport scenarios in order to improve coverage.

This is the motivation for the two new severity measures for transport scenarios we propose: The minimum transport moment and the potential transport moment. In this context, we introduce the Maximum Transportation Problem (MaxTP) and the Maximum Potential Transport Moment Problem (MaxPTM), whose goal is to determine worst-case scenarios corresponding to the respective measures.

MaxTP and MaxPTM can be modeled as Stackelberg games. First, the leader chooses a transport scenario. Then the follower, i.e., the TSO, routes the flow through the network according to a given transport model and objective function, which represents the severity measure. To determine worst-case scenarios, the leader chooses a transport scenario that leads to a maximum transport effort for the TSO. In other words, the goal of the leader is to maximize the objective function value of the follower.

Given a transport scenario selected by the leader, the follower in MaxTP solves the induced Transportation Problem, i.e., the resulting Minimum Cost Flow Problem without capacity restrictions, and we obtain a linear bilevel optimization formulation with interdicting objective functions. To solve MaxTP, we reformulate it as MIP. Therefore, we first introduce the notion of solution-equivalency for MaxTP instances. This concept helps us to identify MaxTP instances whose feasible solutions can easily be mapped onto each other while the objective function values are preserved. In this context, we introduce the so-called L^1 instances, which feature a tripartite and acyclic network. This structure allows us to apply a KKT-reformulation, provide bounds for the corresponding variables, and eventually derive a MIP model by applying the big-M technique.

MaxPTM features a linear potential-based flow model, which can be considered more realistic w.r.t. the physics of gas flow. This model has the advantage that, given a transport scenario, it admits a unique feasible solution w.r.t. the flows. Therefore, and since the potential transport moment exclusively relies on these values, MaxPTM can directly be reformulated as MIP.

A case study based on the `gaslib-582` instance from the GasLib benchmark library [147] demonstrates that our new severity measures should be considered when designing test sets for evaluating the technical capacity of gas transport networks. In particular, the worst-case transport scenarios determined with MaxTP and MaxPTM exceed the corresponding maximum severity measure values from the provided instance set by 23% and 30%, respectively.

How can the transport infrastructure be monitored efficiently?

Malfunctioning hardware does not only affect the control of the entire network but also poses a threat to its safe operation. For example, it is crucial to regularly check the pipelines in order to detect leakages early on. In the case of natural gas networks, this is necessary to prevent methane emissions, while embrittlement is a major concern in hydrogen transport. Thus, the TSOs have to monitor their infrastructure continuously, and it is a natural goal to do this cost-efficiently.

Chapter 5 discusses the idea of using uncrewed aerial vehicles (UAVs) for monitoring the transport infrastructure. To optimize their routing, we introduce and discuss the Length-Constrained Cycle Partition Problem (LCCP), a new generalization of the Traveling Salesperson Problem (TSP). Let the vertices of a given undirected graph represent the objects or areas to monitor, and weights on the edges correspond to the flying times between two adjacent vertices. Furthermore, each vertex is associated with a so-called critical weight value, which represents the maximum duration for which it can be left unattended. A vertex-disjoint cycle partition is a feasible solution for LCCP if the length of each cycle, which represents the flying route of an individual UAV, does not exceed the critical weight value of each of the vertices it contains. We require the flying routes to be disjoint to avoid interferences.

Next, we introduce the Most-Critical-Vertex-Based Heuristic (MCV) for LCCP. Its basic idea is to choose a vertex with minimum critical weight, iteratively select other vertices and extend them to a cycle while the critical weight conditions are satisfied, stop if no further extension seems to be possible, delete the resulting cycle from the graph, and restart the process until all vertices have been assigned. Moreover, we introduce two exact MIP models for LCCP. The first one is based on an adaption of the Miller–Tucker–Zemlin formulation for TSP, and the second one features a variant of subtour elimination constraints.

Moreover, we introduce conflict hypergraphs for LCCP, where a hyperedge represents a set of vertices that cannot be contained in a common cycle. Based on cliques in this graph, we derive valid inequalities and lower bounds for the MIPs.

Finally, by combining MCV, the MIP models, the valid inequalities and lower bounds from the conflict hypergraphs, and additional symmetry-breaking constraints, we determine optimal solutions for problem instances with up to fifty vertices in our TSPLIB- and ATSPLIB-based computational experiments.

1.2. Publications

Parts of this thesis have already been published in international journals or peer-reviewed conference proceedings. Others are discussed in articles that have been submitted and are under review or are currently in preparation. However, nearly all content is publicly available in technical reports. In particular:

- ▷ The majority of the content of Chapter 3, especially on natural gas transport, is based on “Optimal Operation of Transient Gas Transport Networks;

Kai Hoppmann-Baum, Felix Hennings, Ralf Lenz, Uwe Gotzes, Nina Heinecke, Klaus Spreckelsen, and Thorsten Koch; Optimization and Engineering, 22(2):735–781, 2021" [86].

Furthermore, basic approaches and first results regarding hydrogen transport are available in "From Natural Gas towards Hydrogen – A Feasibility Study on Current Transport Network Infrastructure and its Technical Control; Kai Hoppmann-Baum, Felix Hennings, Janina Zittel, Uwe Gotzes, Nina Heinecke, Klaus Spreckelsen, and Thorsten Koch; Technical Report 20-27, ZIB, 2020" [87]. A corresponding paper featuring the updated computational experiments presented in this thesis has been submitted.

The co-authors provided parts of the input data and helped with multiple fruitful discussions regarding the underlying ideas and topics of this chapter. The author of this thesis, who is also the main author of the papers, is responsible for the scientific content.

- MaxTP and the corresponding model and a first solution approach in Chapter 4 have first been introduced in "Finding Maximum Minimum Cost Flows to Evaluate Gas Network Capacities; Kai Hoppmann and Robert Schwarz; Operations Research Proceedings 2017, pages 339–345, 2018" [83].

  The detailed discussions on the Reference Point Method, MaxPTM, and the case study are available in "Using Bilevel Optimization to find Severe Transport Situations in Gas Transmission Networks; Kai Hennig and Robert Schwarz; Technical Report 16-68, ZIB, 2016" [75].

  Both articles have been joint work with Robert Schwarz, who motivated the author to conduct research in this direction in the first place and helped with many valuable suggestions.

  MinTP and all complexity results provided in this chapter have been published in "On the Complexity of Computing Maximum and Minimum Min-Cost-Flows; Kai Hoppmann-Baum; Networks, 79(2):236–248, 2022" [84].

  The main results, i.e., the MIP model and the concepts and results needed to derive it, have been developed by the author of this thesis. A corresponding paper is currently being prepared.

- Chapter 5, which is mainly related to LCCP, is based on two closely related papers. The first one was published as "Minimum Cycle Partition with Length Requirements; Kai Hoppmann, Gioni Mexi, Oleg Burdakov, Carl Johan Casselgren, and Thorsten Koch; Integration of Constraint Programming, Artificial Intelligence, and Operations Research, pages 273–282, 2020" [82].

  The follow-up paper, which generalizes LCCP and features new theoretical results, an improved heuristic algorithm, and a novel MIP model, is available online as "Length-constrained cycle partition with an application to UAV routing*; Kai Hoppmann-Baum, Oleg Burdakov, Gioni Mexi, Carl Johan Casselgren, and Thorsten Koch; Optimization Methods and Software, 2022" [85].

Oleg Burdakov had the original idea regarding LCCP and closely advised the author during his work on it. A first solution approach was developed together with Gioni Mexi in the context of his Bachelor thesis. The discussions with Carl Johan Casselgren and Thorsten Koch helped to put the work into context. The author of this thesis, who is also the main author of both publications, is responsible for the scientific content.

2. Mathematical Prerequisites

This chapter introduces the fundamental mathematical concepts that we apply within the thesis. We start with definitions for mixed-integer and linear programming and discuss important theoretical results as well as main solution approaches for both in Section 2.1. Afterward, we give a brief idea of general multilevel programming and then focus on (mixed-integer) linear bilevel problems in Section 2.2. Besides motivating the latter using an example application, we describe solution methods based on a single-level reformulation and techniques from mixed-integer programming. We conclude this chapter with a short introduction to graphs and (potential) flow networks and define classic related optimization problems in Section 2.3. The motivation here is that gas transport networks are usually modeled as graphs or flow networks, and problems like the Minimum Cost Flow Problem play an essential role throughout this thesis.

2.1. Mixed-Integer Programming

A great majority of the modeling approaches and solution methods presented in this thesis is based on algorithms within which mixed-integer or linear programs must be solved. Therefore, this section states fundamental definitions, results, and algorithms regarding these two problem classes. Most of the terminology and notation is adapted from the PhD theses of Achterberg [1] and Berthold [14].

Given a matrix $A \in \mathbb{R}^{m \times n}$, as well as vectors $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $l \in (\mathbb{R} \cup \{-\infty\})^n$, $u \in (\mathbb{R} \cup \{+\infty\})^n$, and a subset $I \subseteq N := \{1, \dots, n\}$. A *mixed-integer (linear) program* (MIP) is an optimization problem of the form

$$\begin{aligned} \min \quad & c^\intercal x && && (2.1)\\ \text{s.t.} \quad & Ax \leq b && && (2.2)\\ & l \leq x \leq u && && (2.3)\\ & x_j \in \mathbb{Z} && \forall j \in I. && (2.4) \end{aligned}$$

While we call x_j with $j \in N \setminus I$ a *continuous variable*, x_j with $j \in I$ denotes an *integer variable* (2.4). Constraint (2.3), which we consider componentwise, denotes the variable *bounds*. Here, l_j represents a lower and u_j an upper bound on x_j. If we have $0 \leq x_j$ for some variable, we refer to it as *nonnegative*, and if $x_j \in \{0, 1\}$ holds for some integer variable, it is called a *binary* or *decision variable*. The latter term stems from the fact that "yes/no"-decisions are usually modeled using this type of variable. In the following, we denote the index set of the binary variables by $B \subseteq I$.

Next, we discuss condition (2.2), which we consider componentwise as well. In particular, each row with index $i \in \{1, \dots, m\}$ represents a *linear constraint*

$$\sum_{j \in N} a_{ij} x_j \leq b_i. \tag{2.5}$$

In accordance with the "$\leq$"-symbol, b_i is called the *right-hand side* of (2.5). Note that we synonymously call linear constraints *inequalities* in the following.

Additionally, $c^\intercal x$, i.e., (2.1), is the *objective function* of the MIP. Note that it represents a linear function of the x-variables.

A vector $x^* \in \mathbb{R}^n$ is called a *feasible solution* for a MIP if it satisfies (2.2)–(2.4), and it is called *optimal* if it minimizes the objective function value among all feasible solutions, i.e., if $c^\intercal x^* \leq c^\intercal \tilde{x}$ holds for all feasible solutions $\tilde{x}$.

The format of the MIP presented above is very general. For example, a maximization problem can be reformulated as minimization problem by multiplying the objective function coefficients with -1. Similarly, "$\geq$"-constraints can be rewritten as "$\leq$"-constraints by multiplying both sides with -1. Additionally, *equality constraints*, i.e., "$=$"-constraints, can be reformulated using two opposing inequalities.

Note that we will usually include the variable bounds as linear constraints in the constraint matrix except for some special cases, e.g., if we want to emphasize that some class of variables is binary or nonnegative. The latter can actually be assumed w.l.o.g. through replacing each variable with two nonnegative variables, i.e., setting $x = x^+ - x^-$ with $x^+, x^- \geq 0$. Here, x^+ covers all possible nonnegative and x^- all possible nonpositive values after the constraint matrix is extended and adapted accordingly. Finally, by an abuse of notation, we use MIP as an abbreviation for the general concept of mixed-integer linear programming in this thesis as well.

There are several special cases of MIP that we are going to come across in the course of this thesis. In particular, depending on the index sets of the integer and binary variables I and B, we call a MIP

- ▷ a *linear program* (LP) if $I = \emptyset$,
- ▷ an *integer program* (IP) if $I = N$,
- ▷ a *mixed binary program* if $B = I$,
- ▷ and a *binary program* (BP) if $B = I = N$.

Before discussing linear programming in more detail, we briefly introduce a more general class of mathematical programs for the sake of completeness.

A mathematical program that is defined analogous to MIP but where a subset of the constraints and the objective can be arbitrary nonlinear functions we call a *mixed-integer nonlinear program* (MINLP). If $I = \emptyset$, we refer to it as *nonlinear program* (NLP). As we focus on MIP and LP in this thesis, we refer to the survey paper of Burer and Letchford [18], the books of Lee and Leyffer [104] and Belotti et al. [12], as well as the paper of Sahinidis [142] and the references therein for more information on this branch of mathematical programming.

2.1.1. Linear Programming and Duality Theory

An important concept in linear programming is duality. Let us consider an original *(primal) LP* in the form

$$\begin{aligned} \min \ & c^\intercal x \\ \text{s.t.} \ & Ax \leq b \\ & x \geq 0. \end{aligned}$$

Its dual LP is defined as

$$\begin{aligned} \max \ & -b^\intercal u \\ \text{s.t.} \ & A^\intercal u \geq -c \\ & u \geq 0. \end{aligned}$$

We say that u_i with $i \in \{1, \ldots, m\}$ is the *dual variable* associated with the i-th constraint of the primal LP. On the other hand, variable x_j with $j \in \{1, \ldots, n\}$ is the *primal variable* associated with the j-th constraint of the dual LP.

There exists a beautiful and rich theory regarding primal and dual linear programs. For example, if one of the two problems has an optimal solution, so has the other, and the objective function values coincide. We refer to the book of Chvatal [25] for a corresponding proof.

Moreover, the concept of *complementary slackness* describes a relationship between the slackness of a primal/dual constraint, i.e., whether it is binding and holds with equality in the considered solution or not, and the positivity of the associated dual/primal variable. Suppose we are given an optimal solution x^* for the primal LP and an optimal solution u^* for the corresponding dual LP. It holds that

- ▷ if $x_j^* > 0$, then the j-th constraint of the dual is binding,
- ▷ if the i-th constraint of the primal is not binding, then $u_i^* = 0$,
- ▷ if $u_i^* > 0$, then the i-th constraint of the primal is binding, and
- ▷ if the j-th constraint of the dual is not binding, then $x_j^* = 0$.

For corresponding proofs, more information about the duality theory, and a more detailed discussion on linear programming we again refer to the book of Chvatal [25] as well as to the book of Bertsimas and Tsitsiklis [15].

2.1.2. Complexity, Algorithms for LP, and the LP Relaxation

While binary programs, mixed binary programs, integer programs, and mixed integer programs are NP-hard problems, see for example Cook [27], linear programs can be solved in polynomial time, which was first shown by Khatchiyan [91]. One algorithm for LP is the famous simplex method invented by Dantzig [28], which is known to

reliably and efficiently solve linear programs in practice. Thus, it is no surprise that the common algorithmic approach for solving MIPs in practice, which we describe in the upcoming Subsection 2.1.3, relies on repeatedly solving LPs that correspond to subproblems closely related to the original MIP.

To describe this approach, we need one more definition. Consider a MIP as defined above. Relaxing the integrality conditions 2.4, i.e., setting $I = \emptyset$, yields an LP, which is called the *LP relaxation* of the corresponding MIP. Note that all feasible solutions of a MIP are also feasible for its LP relaxation.

2.1.3. LP-Based Branch-and-Bound for MIP

Branch-and-bound, which was first described by Land and Doig [103], is the standard method used in practice to solve MIPs. To the best of our knowledge, all state-of-the-art MIP solvers, e.g., SCIP, Xpress, Gurobi, or CPLEX, use variants of LP-based branch-and-bound, which are additionally enhanced by incorporating several other algorithmic subroutines.

Basic LP-based branch-and-bound can be described as follows. We start with the original MIP and solve its LP relaxation. If the resulting solution $\tilde{x}$ is optimal for the LP relaxation and happens to satisfy the integrality conditions (2.4), we have found an optimal solution for the original MIP. Otherwise, we select an integer variable x_j that takes on some value $\tilde{x}_j \in \mathbb{R} \setminus \mathbb{Z}$ and *branch* on it, i.e., we divide the problem into two subproblems where we add the additional constraint $x_j \leq \lfloor \tilde{x}_j \rfloor$ to one of them and $x_j \geq \lceil \tilde{x}_j \rceil$ to the other. The same procedure is now applied to the two subproblems. There are three cases where we refrain from further branching on a subproblem. First, the LP relaxation of the subproblem is a feasible solution for the original MIP, i.e., the integrality conditions (2.4) are satisfied. Second, the LP relaxation is infeasible. Third, we found a feasible solution for the LP relaxation, but its objective function value is greater than or equal to the objective function value of the current *incumbent*, i.e., the best feasible solution for the MIP that has been found so far, which represents an *upper bound* on the value of an optimal solution. In all of these three cases, it does not make sense to consider further subproblems with additional constraints as these cannot yield better feasible solutions w.r.t. the original MIP.

As mentioned above, several algorithmic procedures can be incorporated into the basic branch-and-bound algorithm to accelerate the solving process. For example, two obvious points for improvement are elaborate selection rules for the next subproblem to process and the integer variable to branch on. For more information regarding this topic, we refer to the PhD thesis of Gamrath [54]. Moreover, it can be beneficial to apply heuristic algorithms to determine feasible solutions and decrease the upper bound during the branch-and-bound process. We refer to the PhD thesis of Berthold [14] for a detailed discussion on this topic. Finally, for an overview of other techniques, e.g., presolving, domain propagation, or cut separation, we refer to the PhD thesis of Achterberg [1].

2.2. Multilevel and Bilevel Optimization

Another important modeling concept used in this thesis is multilevel programming. While we consider a single objective function and a single decision-maker (DM) in MIP, i.e., all variable values are chosen to minimize the objective function, multilevel programming allows us to model sequential decision processes involving multiple DMs. In particular, each DM controls an individual subset of the variables and pursues to an individual objective function and an individual set of constraints. In this context, we assume that the DMs have perfect information and act rationally. This means that the higher levels, i.e., the DMs that act first when selecting values for their respective variables, can predict the reactions of all lower-level DMs. This is because the latter act rationally as well, i.e., according to their individual objective function, their individual set of constraints, the decisions that have already been taken by the higher-level DMs, and in anticipation of the reactions of subsequent lower-level DMs. For more information on multilevel programming, we refer to the book of Migdalas et al. [111]. In the following, we focus on the "simplest" nontrivial case of multilevel programming and consider two DMs, i.e., bilevel programming.

2.2.1. Linear Bilevel Programming

Bilevel programming was first introduced by the German economist Heinrich Freiherr von Stackelberg in his habilitation thesis "Marktform und Gleichgewicht" [158], and it is therefore also known as Stackelberg game. While the upper-level player is commonly referred to as the leader, the lower level is called the follower. Considering a bilevel problem as a game, the leader starts by making the first move while the follower reacts optimally to it w.r.t. its objective. Thus, the leader has to anticipate the follower's response and incorporate it into its considerations to obtain an outcome that is best w.r.t. its objective.

To illustrate this concept using an example, consider the toll pricing problem described in the paper of Labbé et al. [100]. Given a transportation network, the leader represents an authority responsible for setting tolls on a specified subset of connections. The goal is to maximize the collected tolls. On the other hand, the follower, whose task is to route the community of the network users, observes the actions taken by the leader. If the authority calls for too expensive tariffs, the users take alternative routes to avoid the tolls, which results in lower costs for them. Thus, in order to determine an optimal toll policy, the leader has to anticipate the routing behavior of the users, incorporate it into its decisions, and set the tariffs in a way so that it does not pay off for them to switch routes. This problem is modeled as a linear bilevel program in [100], which we are going to discuss shortly.

Next, we introduce fundamental notation and definitions regarding bilevel programming. We focus on mixed-integer linear bilevel problems here, which can be seen as a generalization of MIP explained in Section 2.1. In particular, all constraints and objective functions are linear, and a subset of the variables may have to take on integer values. Thereby, we mainly follow the book of Bard [8].

Given matrices $A_1 \in \mathbb{R}^{m_1 \times n_1}, B_1 \in \mathbb{R}^{m_1 \times n_2}, A_2 \in \mathbb{R}^{m_2 \times n_1}, B_2 \in \mathbb{R}^{m_2 \times n_2}$, vectors $b_1 \in \mathbb{R}^{m_1}$, $b_2 \in \mathbb{R}^{m_2}$, $c_1 \in \mathbb{R}^{n_1}$, $d_1 \in \mathbb{R}^{n_2}$, $c_2 \in \mathbb{R}^{n_1}$, $d_2 \in \mathbb{R}^{n_2}$ as well as index subsets $I_1 \subseteq N_1 := \{1, \dots, n_1\}$ and $I_2 \subseteq N_2 := \{n_1 + 1, \dots, n_1 + n_2\}$. A *mixed-integer (linear) bilevel program* (MIBP) is an optimization problem of the form

$$\min_x \quad c_1^\intercal x + d_1^\intercal y \tag{2.6}$$
$$\text{s.t. } A_1 x + B_1 y \leq b_1 \tag{2.7}$$
$$x \geq 0 \tag{2.8}$$
$$x_j \in \mathbb{Z} \quad \forall j \in I_1 \tag{2.9}$$
$$\min_y \quad c_2^\intercal x + d_2^\intercal y \tag{2.10}$$
$$\text{s.t. } A_2 x + B_2 y \leq b_2 \tag{2.11}$$
$$y \geq 0 \tag{2.12}$$
$$y_j \in \mathbb{Z} \quad \forall j \in I_2. \tag{2.13}$$

Constraints (2.6)–(2.9) describe the *upper-level* or *leader's problem*, which additionally incorporates the *lower-level* or *follower's problem* (2.10)–(2.13). While the leader controls the x-variables, the follower controls the y-variables. The remaining nomenclature is similar to MIP, i.e., (2.6) and (2.10) denote the objective functions and (2.7) and (2.11) denote the linear constraints of the upper and lower level, respectively. Upper-level constraints are synonymously called *coupling constraints* if they explicitly depend on y-variables. Similarly, y-variables contained in coupling constraints are also referred to as *coupling variables*. Finally, the integrality restrictions on the variables regarding index sets $I_1 \subseteq N_1$ and $I_2 \subseteq N_2$ are stated in (2.9) and (2.13), respectively.

Next, a tuple $(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ is called a *feasible solution* for MIBP if it satisfies (2.7)–(2.13). In particular, $y \in \mathbb{R}^{n_2}$ has to be an optimal solution for the MIP (2.10)–(2.13) induced by the selected $x \in \mathbb{R}^{n_1}$. Moreover, (x, y) is called *optimal* if it minimizes the objective function value of the leader among all feasible solutions, i.e., if $c_1^\intercal x + d_1^\intercal y \leq c_1^\intercal \tilde{x} + d_1^\intercal \tilde{y}$ holds for all feasible solutions $(\tilde{x}, \tilde{y})$.

However, the problem above is not well-posed as the optimal solution of the follower does not necessarily have to be unique. Several methods have been proposed in the literature to overcome this and all of them require some assumptions regarding the level of cooperation between the leader and the follower. The most common one and the one we apply in this thesis is *optimistic bilevel programming*. Here, we assume that the leader can influence the follower to choose an optimal solution that is best for the leader w.r.t. its objective function. For the sake of completeness, we additionally want to mention *pessimistic bilevel programming* here, where the follower chooses an optimal solution that is worst for the leader. For more information on pessimistic bilevel programming, we refer to the book chapter of Liu et al. [107]. Finally, we want to emphasize that the leader has to choose the x-variables such that the lower level admits a feasible solution. If this is not possible, the MIBP itself is infeasible.

When developing solution algorithms for MIBP and bilevel programming in general, it is crucial to analyze the degree of coupling between the two levels and the structure and properties of the lower level. Nevertheless, most state-of-the-art algorithms apply techniques originating from MIP, e.g., branch-and-bound and cutting planes. For a survey on these techniques, we refer to Kleinert et al. [93].

In the following, we concentrate on solution approaches for *linear bilevel programs* (BP). BP is the special case of MIBP where $I_1 = I_2 = \emptyset$, i.e., all variables are continuous. Furthermore, we again assume w.l.o.g. that all variables are nonnegative. However, although LP is solvable in polynomial time, BP is NP-hard in the strong sense as shown by Hansen et al. [71]. We additionally recommend the paper of Audet et al. [6], where a nice reduction from mixed binary programming is presented.

There exist several solution algorithms for BP, e.g., the Kth-Best Algorithm or approaches based on parametric programming theory, see the book chapter of Calvete and Galé [21] for more information. In the following, we discuss one of the most common solution approaches, namely to apply a classic Karush–Kuhn–Tucker (KKT) reformulation and convert the BP into a single level optimization problem.

2.2.2. KKT-Conditions-Based Algorithms

The *Karush–Kuhn–Tucker* conditions (KKT) for nonlinear programming are a set of necessary and sufficient optimality conditions, see the original papers of Karush [90] and Kuhn and Tucker [99] in the book by Giorgi and Kjeldsen [61] for details. In the special case of linear programming, they consist of the variables and constraints of the primal and the dual linear program as well as corresponding complementary slackness conditions. Lets assume w.l.o.g. that all variables are nonnegative and that the variable bounds are included in the constraint matrices, i.e., lets consider the primal and dual LP definitions from Section 2.1. We then derive the following nonlinear optimization problem when replacing the lower-level LP with its KKT conditions

$$\min_{u,x,y} \quad c_1^\intercal x + d_1^\intercal y \tag{2.14}$$

$$\text{s.t.} \quad A_1 x + B_1 y \leq b_1 \tag{2.15}$$

$$A_2 x + B_2 y \leq b_2 \tag{2.16}$$

$$-uB_2 \leq d_2 \tag{2.17}$$

$$(d_2 + uB_2)y = 0 \tag{2.18}$$

$$u(b_2 - A_2 x - B_2 y) = 0 \tag{2.19}$$

$$u, x, y \geq 0, \tag{2.20}$$

which is equivalent to (2.6)–(2.13) if $I_1 = I_2 = \emptyset$. Here, y and (2.16) represent the primal variables and constraints of the lower level LP. Furthermore, $u \in \mathbb{R}^{m_2}$ represents the row vector of the corresponding dual nonnegative continuous variables. The corresponding dual constraints are stated in (2.17). Additionally, (2.18) and (2.19) model the complementary slackness conditions of the follower's LP, which

we synonymously call *complementarity constraints*. These nonlinear constraints enforce that a primal/dual inequality is binding or that the corresponding dual/primal variable is zero. Finally, we want to note that the same reformulation can be applied even if $I_1 \neq \emptyset$.

One way to directly solve the NLP above is to apply the *SOS-1 technique*. Thereby, the complementarity conditions are initially omitted and then "branched" on afterward during the solving process. For more details on this method, we refer to the work of Bard and Moore [9].

A second, very popular method is to apply a *big-M reformulation* of the complementarity conditions, which was first described by Fortuny-Amat and McCarl [47]. Here, an auxiliary binary variable $z \in \{0, 1\}$ is introduced for each constraint (2.18) and (2.19). Afterward, the complementarity condition is replaced by two linear constraints. For example, a constraint of the type (2.19) is replaced by

$$b_2 - A_2 x - B_2 y \leq M_p(1 - z) \tag{2.21}$$

$$u \leq M_d z \tag{2.22}$$

where are M_p and M_d are upper bounds on the terms $b_2 - A_2 x - B_2 y$ and u, respectively. Depending on the value of z, one of the two terms is forced to be equal to zero, which implies (2.19). An analogous reformulation can be applied to constraints (2.18). Note that the upper bounds have to be valid to derive a correct MIP model and cannot be chosen arbitrarily, as discussed by Pineda and Morales [132].

Finally, computational experiments conducted by Kleinert and Schmidt [95] show that the big-M approach should be preferred if valid (and small) bounds are available. However, since verifying the correctness of upper bounds is itself an NP-hard problem, see Kleinert et al. [94], it is often beneficial to exploit problem-specific knowledge and the structure of the underlying problem to derive them.

2.3. Graphs, Flow Networks, and Related Problems

Gas transport networks are usually modeled as some kind of (potential-based) flow network. Hence, we introduce basic notation and definitions regarding them in this section. The nomenclature is adapted to align with the terms that are commonly used in gas transport. We recommend the books of Grötschel et al. [66] and Korte and Vygen [98] for an adequate treatment of the topic.

Moreover, we introduce and discuss one of the most fundamental and classic network flow problems, which serves as the basis for many modeling and algorithmic approaches not only in this thesis: The Minimum Cost Flow Problem (MCF). Based on it, we afterward derive definitions for two other well-known problems, namely the Transportation Problem (TP) and the Shortest Path Problem (SP). For more general information regarding flow networks, we refer to the book of Ahuja et al. [4] and the article of Magnanti and Wong [108].

2.3.1. Graph Theory

A *(directed) graph* $G = (\mathcal{V}, \mathcal{A})$ is a tuple consisting of a nonempty and finite set of *vertices* $\mathcal{V} := \{v_1, \dots, v_n\}$ and a finite set $\mathcal{A} \subseteq \mathcal{V} \times \mathcal{V}$ of (ordered pairs of) vertices, which we call *arcs*. We assume that each arc features pairwise different vertices, i.e., we do not consider loops and require that $i \neq j$ holds for each arc $(v_i, v_j) \in \mathcal{A}$. However, we explicitly allow *parallel arcs*, i.e., arcs that have identical start- and end-vertices. If two arcs connect the same two vertices but in opposite directions, we call them *antiparallel*. Finally, while $\delta^+(v) := \{(v, w) \in \mathcal{A} \,|\, w \in \mathcal{V}\} \subseteq \mathcal{A}$ and $\delta^-(v) := \{(w, v) \in \mathcal{A} \,|\, w \in \mathcal{V}\} \subseteq \mathcal{A}$ denote the set of outgoing and ingoing arcs, respectively, $\delta^+(v) \cup \delta^-(v) =: \delta(v) \subseteq \mathcal{A}$ denotes the set of all arcs incident to $v \in \mathcal{V}$.

In contrast, an *undirected graph* $G = (\mathcal{V}, \mathcal{E})$ features a finite set of unordered pairs of vertices $\mathcal{E}$, which we call *edges*. Again, we do not consider loops, and we note that edges can be parallel, but no further distinction is possible here. Accordingly, $\delta(v) := \{\{v, w\} \in \mathcal{E} \,|\, w \in \mathcal{V}\} \subseteq \mathcal{E}$ denotes the set of edges incident to $v \in \mathcal{V}$.

Next, a *(directed) path* in a (directed) graph G, which starts at $v_1 \in \mathcal{V}$ and ends at $v_k \in \mathcal{V}$, is a sequence $(v_1, a_1, v_2, a_2, \dots, a_{k-1}, v_k) =: p_{v_1 v_k}$ where $a_i = (v_i, v_{i+1}) \in \mathcal{A}$ for all $i \in \{1, \dots, k-1\}$ and all nodes in the sequence are distinct, i.e., $v_i \neq v_j$ for $i, j \in \{1, \dots, k\}$ with $i \neq j$. In case that $v_1 = v_k$, the sequence is called a *(directed) cycle*. Corresponding definitions for undirected graphs follow accordingly.

2.3.2. Flow Networks

A *flow network* is a directed graph $G = (\mathcal{V}, \mathcal{A})$ where each arc $a \in \mathcal{A}$ features a nonnegative *capacity* value $c_a \in \mathbb{R}_{\geq 0}$. We synonymously call $\mathcal{V}$ the set of *nodes* in the context of flow networks. Additionally, we are given *boundary values* $b_v \in \mathbb{R}$ for all nodes $v \in \mathcal{V}$. If $b_u > 0$ for some $u \in \mathcal{V}$, then u is called an *entry* and b_u its *supply*. If $b_w < 0$ for some $w \in \mathcal{V}$, then w is called an *exit* and b_w its *demand*. Accordingly, we denote the set of entries by $\mathcal{V}^+ := \{v \in \mathcal{V} \,|\, b_v > 0\} \subseteq \mathcal{V}$ and the set of exits by $\mathcal{V}^- := \{v \in \mathcal{V} \,|\, b_v < 0\} \subseteq \mathcal{V}$. All other nodes $v \in \mathcal{V}^0 := \mathcal{V} \setminus (\mathcal{V}^+ \cup \mathcal{V}^-)$ with $b_v = 0$ are called *inner nodes*. Hence, we denote a flow network as a quadruple $(\mathcal{V}, \mathcal{A}, c, b)$.

Next, a vector $f \in \mathbb{R}^{|\mathcal{A}|}$ is called a *feasible (network) flow* for $(\mathcal{V}, \mathcal{A}, c, b)$ if

$$\sum_{a \in \delta^+(v)} f_a - \sum_{a \in \delta^-(v)} f_a = b_v \qquad \forall v \in \mathcal{V} \tag{2.23}$$

$$0 \leq f_a \leq c_a \qquad \forall a \in \mathcal{A}. \tag{2.24}$$

Here, (2.23) ensures that the supplies and demands of the entries and exits are satisfied and that flow conservation holds at all inner nodes, i.e., the amount of flow entering an inner node must leave it, too. We refer to (2.23) as flow conservation constraints in the following. Additionally, the flow f_a on arc $a \in \mathcal{A}$ is nonnegative and bounded from above by the corresponding capacity c_a (2.24). Note that the sum of the supplies has to equal the sum of the absolute demands to allow for a feasible flow, i.e., supply and demand must be *balanced*.

2.3.3. The Minimum Cost Flow Problem

The Minimum Cost Flow Problem (MCF) is defined on a flow network where we additionally consider a nonnegative *arc length* $\ell_a \in \mathbb{R}_{\geq 0}$ for each arc $a \in \mathcal{A}$. Thus, an instance of MCF is a quintuple $(\mathcal{V}, \mathcal{A}, \ell, c, b)$, and the goal is to find a feasible network flow $f^* \in \mathbb{R}^{|\mathcal{A}|}$ that minimizes the sum of the products of the arc flows and the arc lengths. In other words, MCF can be modeled as the following LP

$$\min_f \sum_{a \in \mathcal{A}} \ell_a f_a \tag{2.25}$$

$$\sum_{a \in \delta^+(v)} f_a - \sum_{a \in \delta^-(v)} f_a = b_v \qquad \forall v \in \mathcal{V} \tag{2.23}$$

$$0 \leq f_a \leq c_a \qquad \forall a \in \mathcal{A}, \tag{2.24}$$

where f_a is the variable representing the flow on the corresponding arc $a \in \mathcal{A}$, and (2.23) and (2.24) are the feasibility conditions concerning the network flow.

Aside from solving the linear programming formulation presented above, several combinatorial algorithms exist based on the rich mathematical theory arising in the context of MCF, see for example the book of Ahuja et al. [4]. Well-known examples are the Minimum Mean Cycle-Cancelling Algorithm of Goldberg and Tarjan [62] or Successive Shortest Path Algorithms, see Edmonds and Karp [41]. For more details on MCF and several example applications, we again recommend [4].

2.3.4. The Transportation Problem

The Transportation Problem (TP) is the special case of MCF where all arcs are uncapacitated, i.e., their capacities are sufficiently large such that they do not have to be considered when determining an optimal solution. For example, if all capacities are greater than or equal to the sum of the supplies, i.e., $c_a \geq \sum_{u \in \mathcal{V}^+} b_u$ for all $a \in \mathcal{A}$, and there exists a feasible solution satisfying the flow conservation constraints (2.23), then there exists an optimal solution satisfying the capacity restrictions (2.24). Thus, we denote a TP instance as quadruple $(\mathcal{V}, \mathcal{A}, \ell, b)$, and considering the LP model for MCF, we derive an LP model for TP by replacing (2.24) with

$$f_a \geq 0 \quad \forall a \in \mathcal{A}. \tag{2.26}$$

2.3.5. Potential-Based Flow Networks

The following definition of (passive) potential-based flow networks, which represents an extension of aforementioned classic flow networks, is based on the paper of Gross et al. [65]. For more details, we additionally refer to the book of Rockafellar [138].

A *potential-based flow network* is a directed graph $G = (\mathcal{V}, \mathcal{A})$ where each node $v \in \mathcal{V}$ is associated with a *potential* $p_v \in \mathbb{R}$ together with lower bounds $\underline{p}_v \in \mathbb{R}$ and upper bounds $\bar{p}_v \in \mathbb{R}$ such that $\underline{p}_v \leq p_v \leq \bar{p}_v$ holds, and each arc $a \in \mathcal{A}$ is associated with a flow q_a with corresponding lower bounds $\underline{q}_a \in \mathbb{R}$ and upper bounds $\bar{q}_a \in \mathbb{R}$

such that $\underline{q}_a \leq q_a \leq \bar{q}_a$ holds. Note that we explicitly allow negative arc flows here, which can be interpreted as flow against the arc's direction. Finally, we are also given a strictly increasing continuous function ϕ with the property that $-\phi(q_a) = \phi(-q_a)$, and an arc-specific parameter β_a for each $a \in \mathcal{A}$.

We call $q \in \mathbb{R}^{|\mathcal{A}|}$ a *feasible (potential-based network) flow* if there exist node potentials $p \in \mathbb{R}^{|\mathcal{V}|}$ such that

$$p_{v_1} - p_{v_2} = \beta_a \phi(q_a) \qquad \forall a = (v_1, v_2) \in \mathcal{A} \tag{2.27}$$

$$\sum_{a \in \delta^+(v)} q_a - \sum_{a \in \delta^-(v)} q_a = b_v \qquad \forall v \in \mathcal{V} \tag{2.28}$$

$$\underline{q}_a \leq q_a \leq \bar{q}_a \qquad \forall a \in \mathcal{A} \tag{2.29}$$

$$\underline{p}_v \leq p_v \leq \bar{p}_v \qquad \forall v \in \mathcal{V}. \tag{2.30}$$

As we can see here, constraint (2.27) couples the flow on an arc with the difference between the potentials of its end-nodes.

In the context of gas transport networks, potentials are usually used to model the squared pressures at the nodes. Further, as differences in the gas pressure induce gas flow, the flow through pipelines typically depends on the potential difference of the corresponding end-nodes. Our model for the transient gas flow through pipelines is for example discussed in the corresponding paragraph in Section 3.5.

2.3.6. The Shortest Path Problem

Recall the definition of a path from Subsection 2.3.1. If we are given additional length values $\ell_a \in \mathcal{A}$ for the arcs, e.g., as in MCF, we define its *length* $\ell(p_{v_1 v_k})$ as the sum of lengths of the arcs it contains, i.e.,

$$\ell(p_{v_1 v_k}) := \sum_{i=1}^{k-1} \ell_{a_i}.$$

Note that the length of a cycle is defined analogously. The goal of the Shortest Path Problem (SP) is to determine a path of minimum length between two given vertices.

SP can also be seen a special case of TP. Suppose we want to find a shortest path from $v_1 \in \mathcal{V}$ to $v_k \in \mathcal{V}$ in a directed graph $G = (\mathcal{V}, \mathcal{A})$ with arc lengths $\ell \in \mathbb{R}^{|\mathcal{A}|}_{\geq 0}$. Consider the induced TP instance with $b_{v_1} = 1$, $b_{v_k} = -1$, and $b_v = 0$ for all $v \in \mathcal{V} \setminus \{v_1, v_k\}$. In an optimal solution, flow will only be sent along shortest paths.

Aside from solving the linear programming formulation of the corresponding TP instance, there are several other famous algorithms for SP. For example, the well-known Dijkstra Algorithm [34] or the Bellman-Ford Algorithm [11, 45], which can even be applied in the case of negative arc lengths.

Finally, for two vertices $v_1, v_k \in \mathcal{V}$, we define the *distance* $d_{v_1 v_k}$ between them as the length of a shortest path $p^*_{v_1 v_k}$, i.e., $d_{v_1 v_k} := \ell(p^*_{v_1 v_k})$. If no path exists, we define $d_{v_1 v_k} := \infty$, and $d_{v_1 v_k} = 0$ holds if $v_1 = v_k$.

3. Optimizing Transient Network Control

Determining a stable control for natural gas transport networks is crucial in order to guarantee the security of the energy supply. Therefore, it is often considered the essential task of the transport system operator (TSO). However, due to the network topologies, which have become large and quite complex over time, the design of the natural gas market, which makes long-term planning and the forecasting of future supplies and demands more complicated, and side effects of green energy production, e.g., gas-fired power plants being ramped up on short notice to make up for a lack of renewable electricity generation, ensuring it has become more difficult.

In this context, after a high-level introduction into the physics and the control of gas transport networks in Section 3.1, and a review of previous and related work in Section 3.2, we present KOMPASS in Section 3.3. KOMPASS is a decision support system for the dispatchers controlling the gas networks. It has been developed in the GasLab of the research campus MODAL [113] in close collaboration with experts and practitioners at Open Grid Europe GmbH (OGE) [120], one of the largest natural gas TSOs in Europe. Like a car's navigation system, KOMPASS continuously recommends measures to ensure a stable network control based on current forecasts of future gas flows and pressures. Thereby, the algorithmic routines that we apply to determine these measures rely on solving mathematical programming models.

Modeling gas transport through networks is a challenging problem mainly due to two aspects: The physics of the transient gas flow in pipelines and the combinatorics behind the setup of compressor stations together with corresponding technical restrictions and limitations. The basic idea implemented in KOMPASS is to split the complexity and handle these two issues in successive stages.

The algorithmic approach applied in the first stage, which represents a main contribution of this thesis, is designed to make important global control decisions, e.g., how to route the flow and where and when to compress the gas. It consists of a tri-level MIP model, followed by a sequential linear programming inspired postprocessing routine. The focus of the approach is on accurately capturing the physics of the transient gas flow through pipelines, while the technical control capabilities of complex subnetworks, e.g., compressor stations, are approximated using the modeling concept of network stations, which we describe in detail in Section 3.4.

The motivation behind the tri-level MIP is to model a hierarchy of different control measures. Thereby, the first and the second level represent so-called non-technical control measures, which can be used to alter the forecasted pressure or flow values, respectively. These values then serve as the basis for the third level, whose goal is to determine a feasible technical network control using as few technical measures as necessary, i.e., to minimize changes in the settings of the network elements. The

latter is considered a main indicator for stability by practitioners. Hence, the overall objective of the formulation is to determine a feasible technical network control with maximum stability while deviations from the predicted flow and pressure values are minimized. The tri-level MIP is explained in detail in Section 3.5, and a solution method based on solving a sequence of closely related single-level MIPs is presented in Section 3.6.

Our model for the transient gas flow in pipelines, which we use in our hierarchical programming formulation, consists of linear constraints. To overcome possible inaccuracies associated with the proposed formulation, we introduce the iterative velocity adjustment procedure (IVAP) as postprocessing routine. This sequential linear programming inspired method aims at deriving physically more accurate results and is the topic of Section 3.7. The question of how the tri-level MIP model and the IVAP are combined is answered in Section 3.8, where a complete description of the first stage algorithm that is used within KOMPASS is given.

To demonstrate the applicability of our approach to natural gas transport, we conduct computational experiments based on a major subnetwork of OGE's infrastructure together with real-world measured pressure and flow data. The setup of these experiments and the results are presented in Section 3.10.

Finally, the first stage's global control decisions are verified in the second stage, where corresponding highly detailed technical control measures for the original network elements are determined. Here, the approach presented by Hennings et al. [78] is used, which we refer to for more information.

Finally, in the context of repurposing the existing natural gas infrastructure, we also investigate whether the converted network can transport the amounts of hydrogen that are necessary to satisfy the energy demands currently covered by natural gas. Therefore, besides explicitly stating our model for hydrogen transport alongside the model for natural gas in Section 3.5, we discuss physical and technical properties that make hydrogen transport challenging and propose a method for converting the natural gas input data into hydrogen scenarios in Section 3.9. The results of the corresponding computational experiments can be found in Section 3.10, too.

A conclusion and an outlook on future research are given in Section 3.11.

3.1. The Physics and Control of Gas Transport Networks in a Nutshell

This section aims at providing some basic knowledge regarding the physics and control of gas transport networks. Therefore, we give a high-level introduction to both topics, focusing on the primary sources of complexity when deriving corresponding mathematical models. The good news is that, although most of the formulas describing the physics seem rather complex, e.g., the Euler equations for the gas flow through pipelines or the power equation for turbo compressor units, the fundamental ideas behind it are not too difficult to understand. For a more detailed introduction to gas transport networks, we refer to the book of Koch et al. [96]. Additionally, we

recommend the book chapter of Hante et al. [72], where sources of complexity arising when modeling the physics and control of gas transport networks are described in detail. Finally, for detailed technical explanations regarding the single network elements, we refer to the book of Cerbe [23].

Pipelines

Gas transport networks mainly consist of cylindric steel pipelines through which the gas can flow. Assuming that all pipelines are horizontal, flow is induced by imbalances in the gas pressure, i.e., the gas flows from areas of high pressure towards areas of low pressure to balance out differences. Thus, a first very general principle when transporting gas from entry to exit points, which are connected via pipelines, is to have higher pressures at the supply and lower pressures at the demand sites.

Typically, the gas flow through pipelines in large-scale networks is described by the Euler equations, a set of nonlinear hyperbolic partial differential equations. Determining a computationally tractable but meaningful formulation for them is one of the main challenges when developing mathematical models for gas transport networks. The Euler equations and our modeling approach for them are discussed in detail in the paragraph regarding pipelines in Subsection 3.5.2.

Valves

Valves are usually installed at all pipeline crossings to control the gas flow's direction and to navigate the desired amounts from the entries towards the exits. A valve has two possible settings: Open or closed. If it is open, gas can flow through it, and the pressures at both ends are identical. On the other hand, if a valve is closed, no gas can flow through it. Further, if the network is disconnected, the pressures in the two resulting subnetworks are decoupled and do not have to coincide anymore. Finally, we remark that there exist several additional operating conditions, e.g., a maximum allowed pressure difference when opening a valve. However, we do not consider those within this thesis.

Compressor Stations

Besides deciding which valve to open or close at which point in time, another class of primary control decisions has to be taken. Due to friction and height differences that may have to be overcome, there is a drop in the pressure when gas flows through the pipelines. This is a problem, especially when gas shall be transported over long distances, as the flow could slow down significantly.

To compensate for this, compressor stations, which comprise multiple compressor units, are installed evenly distributed at appropriate locations all over the network. A compressor unit is a mechanical device that can increase the gas pressure in the direction of the flow. In this thesis, we consider turbo compressors as they represent most of the units installed in OGE's transport network. Roughly speaking, turbo

compressors apply the reverse concept of a turbine and increase the pressure of the gas by mechanically reducing its volume.

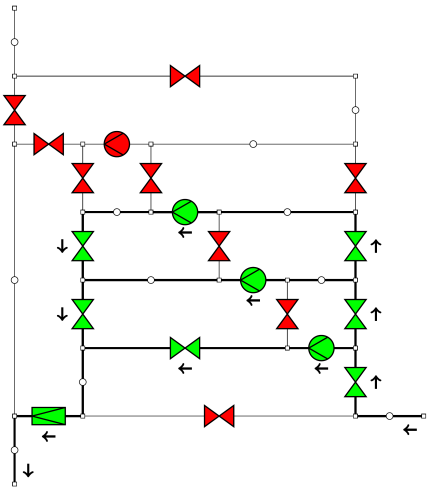
However, there are several physical and technical restrictions associated with this process that limit their capabilities. For example, there is a bound on the maximum flow that can pass through, a maximum compression ratio, which bounds the ratio of the ingoing and outgoing pressure, and a maximum available power. Furthermore, the amount of flow, the pressure ratio, and the power necessary to realize it are additionally coupled through the nonlinear power equation. Similar to the Euler equations, determining a computationally tractable but meaningful formulation for a compressor is another main modeling issue, which we discuss in detail in the paragraph regarding compressor arcs in Subsection 3.5.3.

A compressor station usually comprises multiple compressor units that are connected via a dense grid consisting of short pipelines and valves. By choosing corresponding settings for the surrounding valves, the capabilities of single compressor units can be combined. In particular, units can be operated sequentially, which allows for higher compression ratios, in parallel, which enables a higher flow throughput, or a combination of the two. An example subnetwork with four compressor units running in different configurations is shown in Figure 3.1. Thus, the number of possible configurations increases, sometimes even exponentially, with the number of available compressor units and adds a challenging combinatorial aspect to the problem of determining a stable control for gas transport networks.

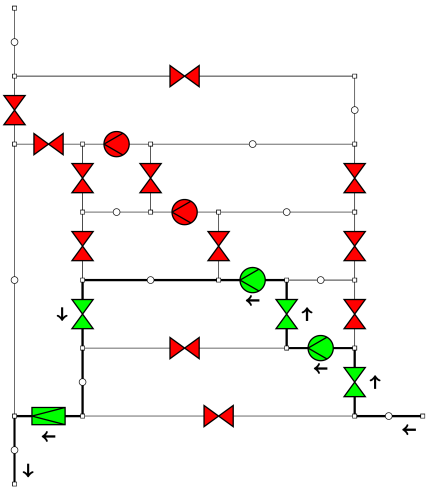
Finally, although we do not consider it in this thesis, for the sake of completeness, we want to mention that the operation of compressor units is often associated with some cost, e.g., for the necessary fuel or power or the negative impact on the environment. Thus, instead of constantly compressing gas, an objective that is often considered in the literature is to figure out when compression is necessary to achieve a feasible network control, i.e., to minimize the total compression cost.

Regulators

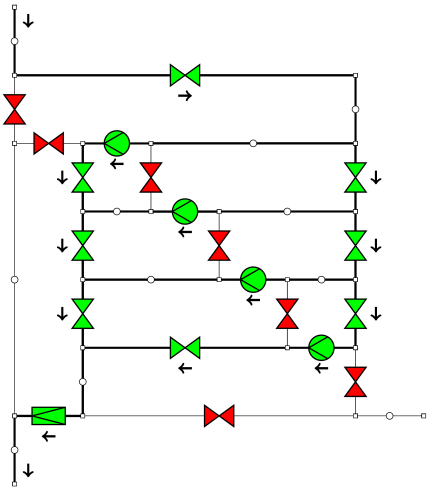
Pipelines are associated with a maximum pressure value that they can withstand. For example, according to Barlow's formula, this upper bound depends on the wall thickness, the outside diameter, and the allowable hoop stress, see Chin et al. [92]. Hence, while pipelines in long-distance transport networks often feature upper bounds of up to 100 bar, the distribution networks of municipalities may only allow for pressures up to eight bar. These bounds have to be respected and taken into account when choosing how to route the flow as well as where to compress the gas and up to which pressures. Additionally, regulators, which are synonymously called control valves, are available at several locations in the network. Besides being fully open or closed, these devices can also be partially open and thereby decrease the pressure in the direction of the flow, which might be necessary to allow for a safe network control. For more, highly detailed information regarding the control of regulators, we refer to the technical report of Hennings et al. [79].



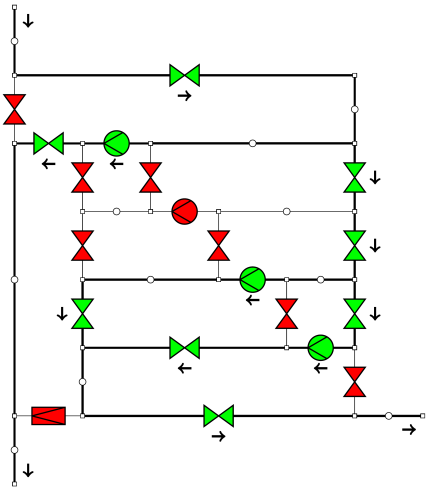
(a) Gas flows from the east to the south and is compressed by three compressor units running in parallel.



(b) Gas flows from the east to the south and is compressed by two compressor units running in serial.



(c) Gas flows from the north to the south and is compressed by four compressor units running in parallel.



(d) Gas flows from the north to the south and the east and is compressed by one and two compressor units running in parallel, respectively.

Figure 3.1.: The four pictures above, taken from Figure 2.10 in Koch et al. [96], show an exemplary subnetwork with four compressor units running in different configurations. The subnetwork contains pipelines —○—, valves —⋈—, a regulator —▭—, and four compressor units —◯—. If an element is highlighted in green, it is open or active. If it is red, it is closed, and there is no flow going through it. Additionally, while gas flows on the thicker black pipelines, there is no gas flow on the thin grey ones.

3.2. Previous and Related Work

There exists plenty of literature regarding operations research and optimization in gas transport, mainly for natural gas. A comprehensive overview of several problem classes can be found in the survey article of Ríos-Mercado and Borraz-Sánchez [141]. Additionally, we recommend the book chapter of Hante et al. [72], where the sources of complexity arising when modeling the physics and control of gas transport networks are described in detail. In the following literature review, we concentrate on the most recent work regarding the optimization of the (transient) control of gas transport networks for the apparent reason.

Most of the research conducted in the past has concentrated on the stationary (or steady-state) gas transport problem. Here, no temporal resolution is considered, and the goal is to determine a feasible network state given the necessary boundary values. Several solution approaches for this problem featuring very detailed models of the network elements are presented in the book of Koch et al. [96], which we also refer to for a general overview of the work related to this problem. Additionally, we want to mention the two more recent papers of Geißler et al. [59, 60] in the context of this problem since the algorithmic approaches introduced in these articles use strategies that are similar to the idea that we will apply. In particular, to solve the basic underlying MINLP model proposed in [60], it is split into two parts, and an alternating direction method is applied. First, pressure and flow values as well as discrete control decisions for all elements outside complex subnetworks, which comprise the compressors, are determined. Afterward, subproblems w.r.t. the omitted subnetworks are solved.

Although it has gained more attention in the last couple of years, research regarding transient (time-dependent) network control is still in its early stages. The first work we discuss here is the PhD thesis of Moritz [114]. In this work, a MIP model is presented where the physics of gas flow through pipelines and the fuel gas consumption of the compressors, a formula closely related to the power equation mentioned in the previous section, are approximated using piecewise linear functions. The goal of the MIP is to determine a network control that minimizes fuel consumption while all supplies and demands are satisfied. The author proposes a branch-and-cut algorithm, which guarantees global optimality w.r.t. the applied approximations. Further, a simulated-annealing-based heuristic is used to generate incumbent solutions for the branch-and-bound processes, see Mahlke et al [109]. Finally, several classes of problem-specific cutting planes are introduced. For example, one is derived from an analysis of the polyhedron induced by minimum run time, minimum downtime, and switching conditions regarding the compressors.

Piecewise-linear approximations for the gas flow through pipelines and fuel gas consumption are also considered in the paper of Domschke et al [36]. Analogous to [114], the goal of their MIP model is to minimize fuel consumption. The authors

also propose an algorithmic approach, where this MIP and a closely related NLP formulation are solved iteratively to reduce the approximation error. Given a solution for the MIP model, the NLP is derived by fixing the binary switching variables for the valves and compressors as well as the compressor powers to the values of the MIP's solution. Additionally, physically more accurate nonlinear models are applied instead of the piecewise-linear approximations. The solution of the NLP is, in turn, used to refine the approximations used in the MIP, which is subsequently solved again. The algorithm terminates when the MIP and the NLP solution coincide.

Burlacu et al. [19] propose a new discretization scheme for the Euler equations. Further, compressors are modeled using lower and upper bounds on the compression ratios and the achievable pressure differences. The resulting MINLP aims at maximizing the amount of gas stored in the network, i.e., the linepack, and an algorithmic approach based on solving a series of MIP models is applied, see Burlacu [20], Geißler et al. [58], and Geißler [57] for details.

Next, Gugat et al. [67] introduce another novel discretization scheme for the Euler equations in their paper. The main idea behind their approach is to apply an instantaneous control, i.e., given a specific discretization of the considered time horizon, they iteratively determine a control for the network considering only the next time step. Thereby, the goal of the resulting MIP models is to deviate as little as possible from the supplies, demands, and pressures of the sources and sinks of the considered time step. The capabilities of the compressors are described as a feasible region using linear constraints.

Finally, several pure NLP models have been proposed in the literature, too. We mention the work of Mak et al. [110] and Zlotnik et al. [167] as representatives here. In both articles, the goal is to decide on compression ratios for the compressors, while fuel consumption shall be minimized. However, an obvious drawback of these models is that, in contrast to the work described above, no discrete decisions are considered, e.g., switching valves or turning on and off compressor station configurations.

3.3. KOMPASS – A Decision Support System for Dispatchers

The dispatchers at OGE control the natural gas transport network mainly based on their personal set of skills, e.g., knowledge from training they receive and their experience. Since they started to face more and more unseen transport scenarios due to the aforementioned reasons, ensuring a stable network control and guaranteeing the security of energy supply has become more complex. Hence, the idea for a decision support system for the transient control of gas transport networks was born: KOMPASS (Kontinuierliches Optimierungs-Modul zur Prognose-Abgesicherten Systemsteuerung/Continuous optimization module for a prognosis-based system control). Its architecture, as implemented and running at OGE, is shown in Figure 3.2.

First, KOMPASS receives the topology of the network, its current state, and prognosis data as input. The latter consists of historic gas flows at the network's entries

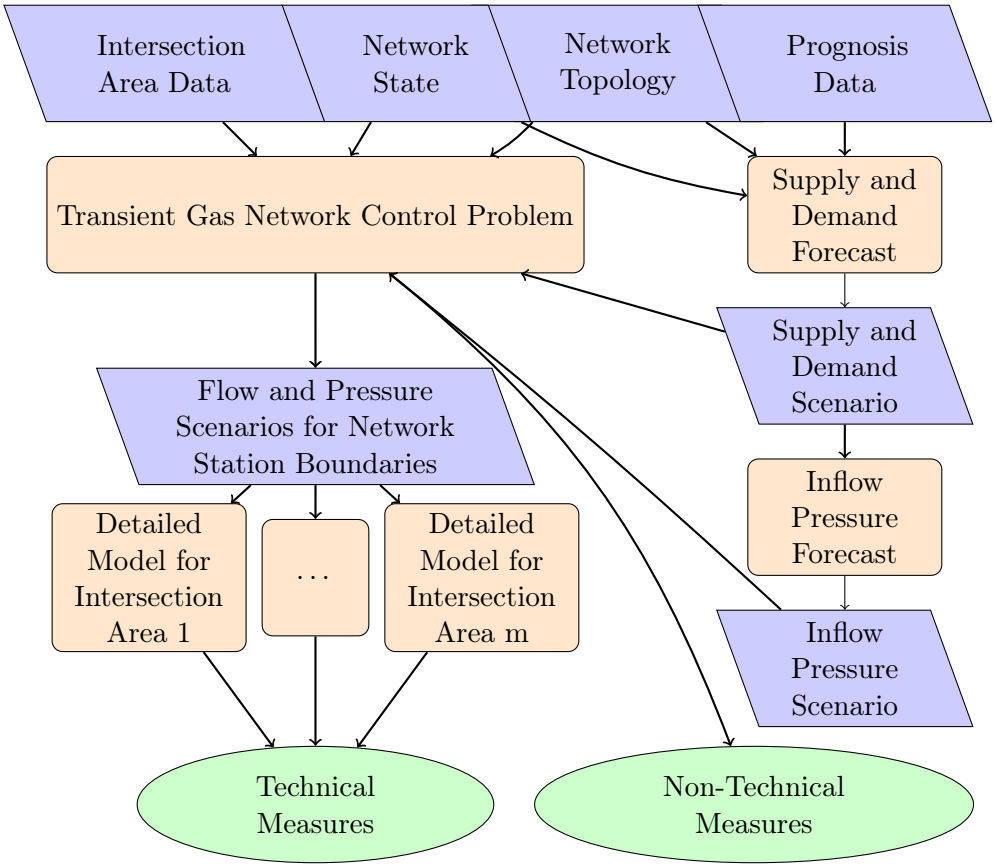


Figure 3.2.: The architecture of KOMPASS and the flow of information within it. Orange rectangles denote computational modules, blue parallelograms describe input data (if in the top row) and intermediate output data (if not in the top row), which serve as input for subsequent algorithmic modules, and green ellipsoids denote output data.

and exits, weather data, and information regarding workdays and holidays. Based on it, future supplies and demands are predicted using the approach of Petkovic et al. [129], and additional time series of future entry pressures are derived heuristically.

As output, highly detailed technical control recommendations shall be determined for all remotely controllable elements during the next couple of hours. To do this, we are given individual characterizations of each element, which are part of the intersection area data that is available for each complex pipeline junction in the network. This data also features a set of operation modes, which, in turn, prescribe settings for all the elements located at that junction and enable particular technical control possibilities. Thus, a central goal of KOMPASS is to suggest an operation mode for each junction and each point in time as well as control recommendations for the single network elements such that the forecasted supplies and demands are satisfied. At the same time, the goal is to maximize the stability of the network control. Switching from one operation mode to another is called a technical measure, and we consider the control of a network to be more stable the less technical measures are applied. We refer to the paper of Hennings et al. [78] for more details regarding the characterizations of the single network elements and the operation modes.

In the real world, the dispatchers try to control the network using technical measures only. However, since this is not always possible, they additionally have some non-technical measures at hand. The most common and standardized ones result in changes to the future supplies and demands by, for example, using contractual options like the interruption of customers or by buying or selling gas, i.e., the usage of so-called balancing energy. For more details on the latter, we refer to Section 4.1, where the German natural gas market is explained in detail. If changing the supplies and demands alone does not enable a feasible network control, the last option is to ask neighboring TSOs for changes in the future entry pressures. In practice, this is done by phone calls, and it can therefore be seen as an emergency and non-standardized option. All this establishes a natural hierarchy on the measures: If the network can be controlled by technical measures only, i.e., without applying non-technical measures, this is most favorable. As a second option, deviations from the supplies and demands are allowed. Finally, if there is still no feasible control for the network, the last resort is to change future entry pressures.

Hence, given the network topology, its current state, time series on supplies and demands as well as on the entry pressures, and the intersection area data, we need to solve a transient gas network control problem on a large-scale and complex real-world gas transport network in KOMPASS as the next step. A corresponding model has to incorporate the operation modes, which comprise the complex combinatorics of the setup of compressor stations, a formulation for the transient gas flow through pipelines, which should be as accurate as possible, and the hierarchy of technical and non-technical control measures.

Due to the nature of the application for which KOMPASS is designed, run time plays a crucial role, too. In particular, the decision support system is supposed to run 24/7 and continuously provide technical and non-technical control recommendations within short time intervals. However, preliminary experiments showed that models

incorporating all these needs simultaneously were computationally intractable or not solvable within reasonable amounts of time. Thus, we decided to split the complexity and pursued a two-stage approach.

In the first stage, we solve a transient control problem where hand-tailored simplified models for each complex pipeline intersection area are applied. In particular, the original junctions are replaced by simplified graph representations, which we call network stations. Their description is also part of the intersection area data, and their derivation is explained in more detail in Section 3.4.

The main idea of the model used here is to conceptually consider the network as network stations, entries, and exits connected by segments of pipelines. The task is to determine important global control decisions for each network station and each point in time, e.g., how to route the flow as well as where and when to compress the gas, while we apply a physically detailed model for the resulting transient gas flows between them. Additionally, the non-technical control measures proposed by KOMPASS are based on the results of the first stage, too.

In the second stage, the flow and pressure values at the boundaries of the network stations from the first stage serve as input for highly detailed models for the original complex pipeline intersection areas. These models are used to verify whether actual operation modes exist that realize the pressure and flow scenarios determined by the first stage. Thereby, stationary formulations focusing on the combinatorics and the technical restrictions of the compressor stations are solved in a first step. The rationale behind this is that intersection areas contain only pipelines of short length, which cannot store or provide much gas for future usage, i.e., they do not feature much linepack. Therefore the transient aspect of the gas flow is neglected at first. However, this aspect is again included in a second step, where a corresponding mathematical model is solved using a rolling horizon approach. For more details on the second stage of KOMPASS, we refer to Hennings et al. [78].

It is important to note that we try to avoid introducing non-linear constraints within our modeling strategy since we aimed at using sophisticated and powerful state-of-the-art MIP and LP solvers as underlying algorithmic routines. However, we will discuss whenever a non-linear formulation may be more exact w.r.t. the physics and the control of gas transport networks in the following.

As it represents a main contribution of this thesis, in the remainder of this chapter, we concentrate on the first stage of KOMPASS, i.e., formulating and solving the transient gas network control problem using the network station model. Next, we explain the network station modeling concept and introduce a tri-level MIP formulation in the upcoming Section 3.4 and Section 3.5, respectively. The latter incorporates the hierarchy regarding the non-technical measures and the technical network control and the rationale behind using hierarchical programming for it is sketched in Figure 3.3.

Nevertheless, for the sake of comprehensibility, we give a written bottom-to-top explanation here, too. The third level features the technical control problem, which tries to maximize the stability of the network by minimizing changes in the settings of the network elements in the network stations, i.e., by minimizing the usage of

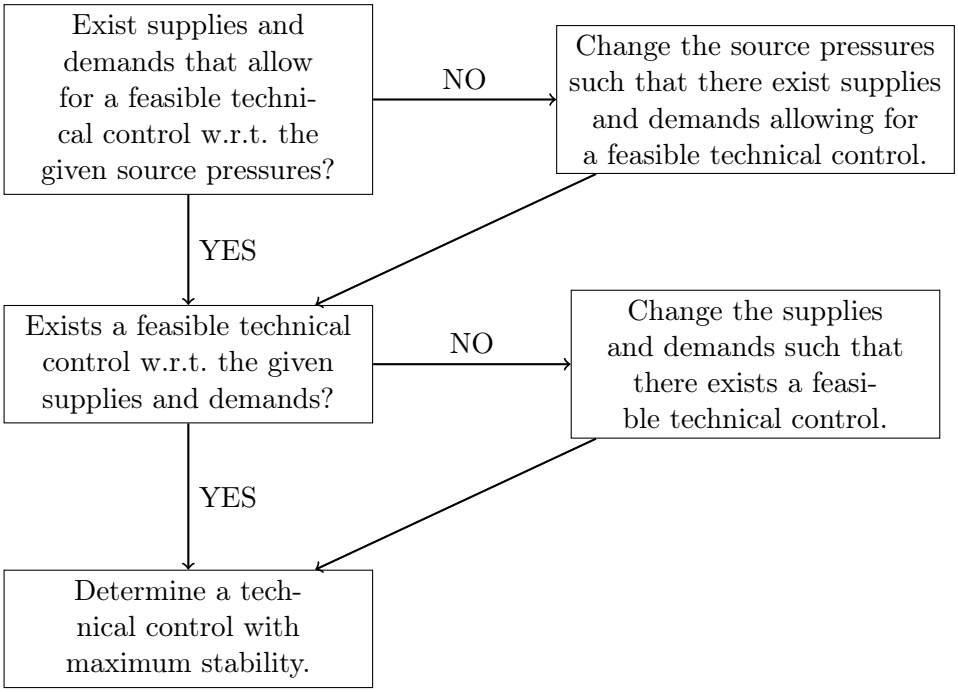


Figure 3.3.: Top-to-bottom explanation of the rationale behind our tri-level formulation. The change instructions are executed such that the sums of the absolute deviations from the input parameters are minimized.

technical measures. The second level minimizes the changes of supplies and demands w.r.t. the sum of absolute deviations necessary to guarantee the feasibility of the third level. The first level pursues a similar goal but minimizes the sum of deviations from future pressure values at the entries instead. Thereby, it takes all possible actions of the second level into account. Hence, the first and the second level minimize the extent of the corresponding non-technical measures in hierarchical order to ensure that a feasible technical control of the network is possible.

3.4. The Network Station Modeling Concept

Most of the elements in gas transport networks whose behavior can be controlled remotely by the dispatchers, such as compressor stations, regulators, or valves, are located at the intersections of major transportation pipelines. The locations of the seven main network stations of OGE's subnetwork, which we consider in our computational experiments in Section 3.10, are shown in Figure 3.4. For each of these junctions and each point in time, exactly one operation mode has to be in use. However, there is a vast number of them because the operation modes comprise, among other things, the combinatorics of the setup of the compressor stations. Thus, due to the induced computational complexity, together with the experts at OGE we developed a hand-tailored simplified graph representation called network station to summarize and approximate the technical control capabilities. The network station model for the junction in the northwest, for example, is visualized in Figure 3.5. While detailed mathematical formulations and corresponding explanations regarding the network station model can be found in Section 3.5, we explain the basic idea behind it and its derivation process here.

Although we are currently working on an automatized process, the derivation of the network station models is currently done manually by experts at OGE, who know the network, its elements, and their control very well. Thus, we give some intuition of how they are created using examples in the following.

First, the intersection areas are identified as connected subgraphs of the network. Their topologies are chosen with the goal in mind to include as many remotely controllable elements as possible while only a few pipelines of significant length are contained. The latter follows the idea behind the two-stage approach in KOMPASS, where the transient gas flow through pipelines is initially considered less important in the second stage, see Section 3.3 for details.

The nodes at the boundaries of these subgraphs are called fence nodes. Additionally, if a subset of the fence nodes features the same behavior, e.g., all are connected to pipelines of large diameter, which run in parallel and nearly always possess the same pressure level and the same direction and amount of flow, they are merged in the network station topology and called a fence group.

Next, we create the topology of the network station. Therefore, we remove the interior of the subgraph and add artificial nodes together with artificial arcs, which connect them and the fence nodes. There are four types of artificial arcs: Shortcuts,

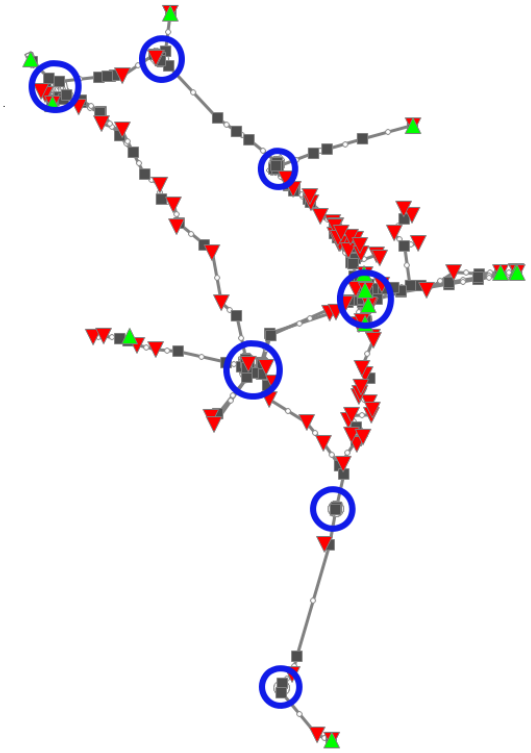
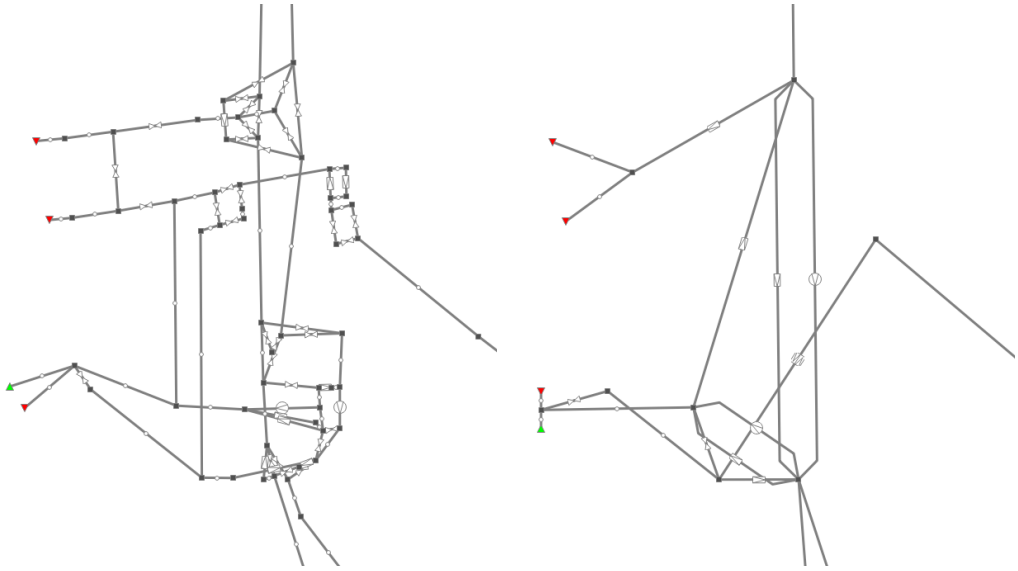


Figure 3.4.: Subnetwork of OGE which we use in our computational experiments. Inner nodes are shown as black squares, entries as green triangles ▲, and exits as red triangles ▼. Furthermore, the seven main network stations are located within the blue circles. All connections visible in this picture are pipelines.

which can be seen as the equivalents of valves, regulating arcs, which can be seen as regulators, compressor arcs, which capture the pressure-increasing capabilities of compressor stations, and combined arcs, which can work as either regulating or compressor arcs. Besides mono-directed arcs, which only support flow going into the direction of their topological orientation, there exists a bi-directed version for each arc type. Here the gas can flow and the mentioned capabilities can be applied in both directions. Shortcuts are always bi-directed by definition.

Artificial nodes are used to decrease the number of necessary artificial arcs and to improve the comprehensibility of the network station model. On the other hand, we can often directly identify artificial arcs with remotely controllable elements in the original topology. Looking at the example in Figure 3.5, we can, for example, identify the two anti-parallel regulators in the northeast of the original network topology with the bi-directed regulating arc in the station model. Additionally, we see two compressing arcs in the network station model, which directly correspond to the two compressor stations depicted in the original topology. Here, the experts



(a) An original pipeline intersection area (b) Corresponding network station

Figure 3.5.: On the left-hand side, an original pipeline intersection area of the network from Figure 3.4 is shown, while the corresponding network station model is depicted on the right. The colored triangles represent entries ▲ and exits ▼, which are located close to the actual network station visualized here. The other network elements here are pipelines —○—, valves/shortcuts —⋈—, regulators/regulating arcs —▷—, compressor stations/compressing arcs —◯—, and bi-directed regulating arcs ⇄.

know that the compressor station in the east is used to compress gas coming from the north and leaving to the south, while the other compressor station can be used to compress gas coming from the south and leaving to the west. This explains the choice of endnodes and the direction of the artificial arcs. However, it is important to note that such a mapping is, in general, not possible. Combined arcs, for example, usually comprise at least one compressor station and one regulator.

Finally, the experts look at the operation modes and create the sets of flow directions and simple states. The former describe possible general directions of flow through the network station. Thus, each flow direction consists of two subsets of fence nodes: Entries, where the gas enters the station, and exits, where it leaves it. On the other hand, a simple state describes a general state of the network station w.r.t. the technical capabilities being in use. In particular, it features of a subset of flow directions that it supports and two subsets of artificial arcs: Arcs that have to be used and arcs that cannot be used. While an unusable arc can conceptually be seen as a closed valve, the former must be used according to the corresponding models described in Subsection 3.5.3. When designing these two sets, the goal is to summarize and approximate the technical capabilities of the operation modes of the original intersection area while keeping the corresponding cardinalities small.

Creating the flow directions and simple states is mainly based on experience and data on which operation modes of the corresponding intersection area have been used in the past. For example, typical flow directions in Figure 3.5 are situations where gas enters from the north or east and leaves to the south or west. However, it can also enter from the south and leaves to the west.

3.5. A Tri-Level MIP for the Stable Transient Control of Gas Networks

Next, we introduce our tri-level MIP model for determining a stable transient control for gas transport networks. Therefore, we present the entities of our network model together with corresponding parameters and introduce variables and constraints capturing the principles of their operation and interplay. Although it can be adapted to other gases, we explicitly state our model for natural gas and hydrogen.

Beforehand, for the sake of comprehensibility, we give a brief guideline w.r.t. the model's structure. While the first level controls the inflow pressure slack variables, the second is in charge of the boundary flow slack variables. These variables, introduced and discussed in Subsection 3.5.2, are used to model the two non-technical measures we consider, i.e., deviations from future entry pressures and the supplies and demands, respectively. The goal of both levels is to minimize the extent of their usage, i.e., the sums of absolute deviations. On the other hand, all constraints belong to the third level, i.e., the level responsible for the technical control, which controls all remaining variables, too. We refer to Section 3.5.5 for more details on the overall tri-level MIP structure and the objective functions.

3.5.1. Basic Notation and Conceptual Model View

In the following, we model a gas transport network as a directed graph $G = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{A}$ the set of arcs.

Time Steps and Granularity

Additionally, we consider a set of time steps $\mathcal{T}_0 := \{0, \dots, k\}$ together with a monotonically increasing function $\tau : \mathcal{T}_0 \to \mathbb{N}$, which represents the granularity. W.l.o.g. we assume that $\tau(0) = 0$ and $\tau(t)$ stands for the number of seconds that have passed until time step $t \in \mathcal{T}_0$ w.r.t. time step 0. Furthermore, we define $\mathcal{T} := \mathcal{T}_0 \setminus \{0\}$ as the set of future time steps for notational purposes.

Node Sets

The node set $\mathcal{V}$ can be partitioned into the set of entries $\mathcal{V}^+$, the set of exits $\mathcal{V}^-$, and the set of inner nodes $\mathcal{V}^0$. Additionally, we call the union of the entries and exits boundary nodes and denote it by $\mathcal{V}^{\mathrm{b}} := \mathcal{V}^+ \dot\cup \mathcal{V}^-$.

Arc Sets

The set of arc can be partitioned into pipelines $\mathcal{A}^{\mathrm{pi}}$ and so-called artificial arcs $\mathcal{A}^{\mathrm{ar}}$, i.e., $\mathcal{A} = \mathcal{A}^{\mathrm{pi}} \dot\cup \mathcal{A}^{\mathrm{ar}}$. The latter are used to control the gas flow and can further be split into four disjoint subsets $\mathcal{A}^{\mathrm{ar}} = \mathcal{A}^{\mathrm{sc}} \dot\cup \mathcal{A}^{\mathrm{rg}} \dot\cup \mathcal{A}^{\mathrm{co}} \dot\cup \mathcal{A}^{\mathrm{cb}}$. Here, $\mathcal{A}^{\mathrm{sc}}$ denotes the set of shortcuts, $\mathcal{A}^{\mathrm{rg}}$ the set of regulating arcs, $\mathcal{A}^{\mathrm{co}}$ the set of compressor arcs, and $\mathcal{A}^{\mathrm{cb}}$ the set of combined arcs. Additionally, while for some of these arcs the corresponding technical control capabilities can only be applied in forward direction, i.e., mono-directed arcs, there also exist bi-directed arcs $\mathcal{A}^{\mathrm{bi}} \subseteq \mathcal{A}^{\mathrm{ar}}$, which can be used in both directions.

Conceptual Network View

Following the idea that was first mentioned in Section 3.3, we conceptually divide the network into two parts. In particular, we consider $m \in \mathbb{N}$ disjoint subgraphs $G_i = (\mathcal{V}_i, \mathcal{A}_i^{\mathrm{ar}})$ with $i \in I := \{1, \dots, m\}$, which we call network stations and are mainly located at major pipeline crossings. Each artificial arc is contained in exactly one of these network stations, i.e., $\dot\bigcup_{i \in I} \mathcal{A}_i^{\mathrm{ar}} = \mathcal{A}^{\mathrm{ar}}$, and no pipelines are present, i.e., $(\bigcup_{i \in I} \mathcal{A}_i^{\mathrm{ar}}) \cap \mathcal{A}^{\mathrm{pi}} = \emptyset$. Moreover, no boundary nodes are contained in a network station and we therefore have $\bigcup_{i \in I} \mathcal{V}_i \subseteq \mathcal{V}^0$.

On the other hand, we call the subgraph induced by the set of pipelines, consisting of $\mathcal{V}^{\mathrm{pi}} := \{v \in \mathcal{V} \mid \delta(v) \cap \mathcal{A}^{\mathrm{pi}} \neq \emptyset\}$, i.e., the set of nodes incident to a pipeline, and $\mathcal{A}^{\mathrm{pi}}$, the connecting network as it connects the network stations, the entries, and the exits with each other.

After stating a mathematical formulation for the connecting network in Subsection 3.5.2, we turn to the network station model in Subsection 3.5.3 and finally discuss constraints that connect both in Subsection 3.5.4.

3.5.2. Connecting Network

This subsection discusses our model for the connecting network, i.e., the subgraph induced by the set of pipelines.

Pressures and Technical Pressure Bounds

For each node $v \in \mathcal{V}^{\mathrm{pi}}$, we are given a nonnegative initial pressure value $p_{v,0} \in \mathbb{R}_{\geq 0}$. Further, we introduce pressure variables $p_{v,t} \in [\underline{p}_{v,t}, \bar{p}_{v,t}] \subseteq \mathbb{R}_{\geq 0}$ for all $t \in \mathcal{T}$. The lower and upper bounds $\underline{p}_{v,t}, \bar{p}_{v,t} \in \mathbb{R}_{\geq 0}$ are called technical pressure bounds and must be respected as their violation can lead to severe damage to the adjacent pipelines.

Supplies, Demands, and Boundary Flow Slacks

For each boundary node $v \in \mathcal{V}^{\mathrm{b}}$ and each time step $t \in \mathcal{T}$ we are given a boundary value $D_{v,t} \in \mathbb{R}$. These values represent future requirements in terms of supply, if $D_{v,t} \in \mathbb{R}_{\geq 0}$ and $v \in \mathcal{V}^{+}$, or demand, if $D_{v,t} \in \mathbb{R}_{\leq 0}$ and $v \in \mathcal{V}^{-}$. However, the boundary values may be adjusted to ensure feasibility of the third level, i.e., the existence of a feasible technical control. Therefore, for each boundary node $v \in \mathcal{V}^{\mathrm{b}}$ and each $t \in \mathcal{T}$ we introduce two continuous variables $\sigma^{d+}_{v,t}, \sigma^{d-}_{v,t} \in \mathbb{R}_{\geq 0}$. The boundary values that are then actually considered in the third level are established through additional variables $d_{v,t} \in \mathbb{R}_{\geq 0}$ for each entry $v \in \mathcal{V}^{+}$, $d_{v,t} \in \mathbb{R}_{\leq 0}$ for each exit $v \in \mathcal{V}^{-}$, and constraints

$$d_{v,t} + \sigma^{d+}_{v,t} - \sigma^{d-}_{v,t} = D_{v,t} \quad \forall v \in \mathcal{V}^{\mathrm{b}}, \forall t \in \mathcal{T}. \tag{3.1}$$

We call $\sigma^{d+}_{v,t}$ and $\sigma^{d-}_{v,t}$ boundary flow slack variables, and they are used to model the non-technical control measure of deviating from given supplies and demands. They are controlled by the second level of our tri-level MIP, whose goal is to minimize their sum, see Section 3.5.5 on the objective functions and the complete model.

Inflow Pressure Bounds and Slacks

Next, for each entry $v \in \mathcal{V}^{+}$ and each point in time $t \in \mathcal{T}$ we are additionally given inflow pressure bounds $\underline{p}^{\mathrm{act}}_{v,t} \in \mathbb{R}_{\geq 0}$ and $\bar{p}^{\mathrm{act}}_{v,t} \in \mathbb{R}_{\geq 0}$ if v has nonzero supply. These bounds, which are tighter than the technical pressure bounds, model the forecasted entry pressures. However, in contrast to the technical pressure bounds, they can be relaxed to ensure feasibility of the third level, i.e., the existence of a feasible technical control. Therefore, we introduce two continuous variables $\sigma^{p+}_{v,t} \in [0, \underline{p}^{\mathrm{act}}_{v,t} - \underline{p}_{v,t}]$ and $\sigma^{p-}_{v,t} \in [0, \bar{p}_{v,t} - \bar{p}^{\mathrm{act}}_{v,t}]$, and constraints

$$p_{v,t} + \sigma^{p-}_{v,t} \geq \underline{p}^{\mathrm{act}}_{v,t} \quad \forall v \in \mathcal{V}^{+} \text{ with } D_{v,t} \neq 0, \forall t \in \mathcal{T}, \tag{3.2}$$

$$p_{v,t} - \sigma^{p+}_{v,t} \leq \bar{p}^{\mathrm{act}}_{v,t} \quad \forall v \in \mathcal{V}^{+} \text{ with } D_{v,t} \neq 0, \forall t \in \mathcal{T}. \tag{3.3}$$

We call $\sigma^{p+}_{v,t}$ and $\sigma^{p-}_{v,t}$ inflow pressure slack variables, and they are used to model the non-technical control measure of deviating even further from the forecasted entry pressures. They are controlled by the first level of our tri-level MIP, whose goal is to minimize their sum, see Section 3.5.5 on the objective functions and the complete model. Note that constraints (3.2) and (3.3) depend on $D_{v,t}$ and not on $d_{v,t}$ as the converse would lead to the introduction of an additional class of binary variables and corresponding constraints.

Pipelines

One-dimensional gas flow in cylindric pipelines is usually described by the Euler equations, a set of nonlinear hyperbolic partial differential equations, see Osiadacz [123]. In this thesis, we assume isothermality, i.e., that the gas temperature remains constant. In that case, these are reduced to the Continuity Equation and the Momentum Equation. While the former ensures conservation of mass, the latter describes the interaction between the force acting on the gas particles and the rate of change in their momentum. For a pipeline $a = (\ell, r) \in \mathcal{A}^{\mathrm{pi}}$ they can be stated as

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial (\rho v)}{\partial x} &= 0 \\ \frac{\partial (\rho v)}{\partial t} + \frac{\partial p}{\partial x} + \frac{\partial (\rho v^2)}{\partial x} + \frac{\lambda_a}{2D_a} |v| v \rho + g s_a \rho &= 0. \end{aligned}$$

The x-variable represents the position in the pipeline w.r.t. the distance from ℓ. Furthermore, t denotes the time, and ρ and v the density and the velocity of the gas, respectively. Additionally, D_a denotes the diameter of the pipeline and the gravitational acceleration is given by g. Further, by λ_a we denote the friction factor of the pipe, which we derive from the formula of Nikuradse. The latter depends on two characteristics of the pipeline only, namely its diameter and integral roughness, see Fügenschuh et al. [50] and Nikuradse [118] for details. Finally, the slope of the pipeline is given by $s_a = \frac{h_r - h_\ell}{L_a} \in [-1, 1]$, where h_ℓ and h_r denote the altitude at ℓ and r.

Next, we reformulate these equations w.r.t. the quantities we are interested in, i.e., mass flow q, pressure p, and the gas velocity v. Mass flow is defined as

$$q = A_a \rho v, \tag{Q}$$

where $A_a = D_a^2 \frac{\pi}{4}$ denotes the cross-sectional area of the pipe. Second, we apply the equation of state for real gases, which describes the relation between the gas pressure p and its density ρ

$$p = \rho R_s T z_a.$$

Here, R_s denotes the specific gas constant and z_a is the compressibility factor of the gas in the pipeline. In the following, we assume both of these values to be constant over time. For the former, this is a consequence of the fact that we assume the molar

mass of the considered gases to be constant. And for the latter, this is a common assumption in natural gas transport, see for example Osiadacz [123]. Therefore, we define z_a as the average of the compressibility factors at both endnodes using the initial pressure values $p_{\ell,0}$ and $p_{r,0}$. Thereby, we apply the approximation formula of Papay [126] for natural gas. For hydrogen, we apply a linear function, which is the result of a linear regression for a set of empirically measured values. In particular, for $a = (l, r) \in \mathcal{A}^{\mathrm{pi}}$ we define

$$z_a := \frac{\alpha(p_{l,0} + p_{r,0})}{2} + \beta, \tag{Z}$$

where $p_{l,0}$ and $p_{r,0}$ are the initial pressures, $\alpha = 6.35882 \cdot 10^{-4}$, and $\beta = 0.99911$.

In a next step, we drop the first and the third summand in the Momentum Equation, since their contribution under typical operating conditions in natural gas transport networks is negligible, see Hennings [77]. Putting this together, we can rewrite the equations and derive the so-called friction dominated model

$$\begin{aligned} \frac{\partial p}{\partial t} + \frac{R_s T z_a}{A_a} \frac{\partial q}{\partial x} &= 0 \\ \frac{\partial p}{\partial x} + \frac{\lambda_a R_s T z_a}{2 D_a A_a^2} \frac{|q|q}{p} + \frac{g s_a}{R_s T z_a} p &= 0. \end{aligned}$$

Next, we discretize these equations using the implicit box scheme proposed by Domschke et al. [36] and Kolb et al. [97]. Here, the length of the pipeline L_a serves as spatial domain while we use the set of time steps $\mathcal{T}_0$ as temporal domain. Thus, for each time step $t \in \mathcal{T}$ we introduce two continuous variables $q_{\ell,a,t}, q_{r,a,t} \in [-\bar{q}_{a,t}, \bar{q}_{a,t}]$, which represent the mass flow into a at ℓ and out of a at r. Note that negative variable values represent mass flow out of a at ℓ and into a at r, respectively, and that $\bar{q}_{a,t}$ denotes a practically reasonable bound on the mass flow. Moreover, for time step $t = 0$ we have fixed initial mass flow values, which we denote as $q_{\ell,a,0}$ and $q_{r,a,0}$. Using these variables and parameters, the discretized equations for two adjacent time steps $t-1$ and t can then be written as

$$\begin{aligned} &\frac{2 R_s T z_a (\tau(t) - \tau(t-1))}{L_a A_a} (q_{r,a,t} - q_{\ell,a,t}) \\ &\qquad + p_{\ell,t} + p_{r,t} - p_{\ell,t-1} - p_{r,t-1} = 0 \qquad \text{(C)} \\ p_{r,t} - p_{\ell,t} + \frac{\lambda_a R_s T z_a L_a}{4 D_a A_a^2} &\left(\frac{|q_{\ell,a,t}| q_{\ell,a,t}}{p_{\ell,t}} + \frac{|q_{r,a,t}| q_{r,a,t}}{p_{r,t}} \right) \\ &\qquad + \frac{g s_a L_a}{2 R_s T z_a} (p_{\ell,t} + p_{r,t}) = 0. \qquad \text{(M)} \end{aligned}$$

Finally, we apply the linear model for the Momentum Equation proposed by Hennings [76]. To derive it, we fix the absolute velocities in the friction-based pressure difference term of the Momentum Equation, i.e., in the third summand, to the absolute gas velocities of the initial time step. Thereby, we derive the following

equations, which we use in our tri-level MIP to model transient gas flow in pipelines

$$\frac{2R_sTz_a(\tau(t)-\tau(t-1))}{L_aA_a}\left(q_{r,a,t}-q_{\ell,a,t}\right) + p_{\ell,t}+p_{r,t}-p_{\ell,t-1}-p_{r,t-1}=0 \quad \forall t\in\mathcal{T} \tag{3.4}$$

$$p_{r,t}-p_{\ell,t}+\frac{\lambda_aL_a}{4D_aA_a}\left(|v_{\ell,0}|\,q_{\ell,a,t}+|v_{r,0}|\,q_{r,a,t}\right) + \frac{gs_aL_a}{2R_sTz_a}\left(p_{\ell,t}+p_{r,t}\right)=0 \quad \forall t\in\mathcal{T}. \tag{3.5}$$

A possible issue that may arise here is that if the velocity of the mass flow increases or decreases significantly over time, we might underestimate or overestimate the friction loss, respectively. An analysis of historic real-world data by Hennings [76] shows that the resulting deviation can be significant, but is negligible in most cases when considering, for example, a time horizon of 12 hours. Nevertheless, Section 3.7 introduces and discusses our iterative velocity adjustment procedure, which is inspired by sequential linear programming. We apply it to determine solutions that are feasible for the nonlinear Momentum Equation (M).

3.5.3. Network Stations

The idea behind the network station model and the process of its derivation are discussed in Section 3.3 and Section 3.4, respectively. Formally, within G there exist $\nu\in\mathbb{N}$ subgraphs $G_i=(\mathcal{V}_i,\mathcal{A}_i^{\mathrm{ar}})$ called network stations, which consist of inner nodes and artificial arcs only, i.e., $\mathcal{V}_i\subseteq\mathcal{V}^0$ and $\mathcal{A}_i^{\mathrm{ar}}\subseteq\mathcal{A}^{\mathrm{ar}}$ for all $i\in\{1,\dots,\nu\}$. Each artificial arc is contained in exactly one network station and each inner node is contained in at most one network station, i.e., $\mathcal{A}_i^{\mathrm{ar}}\cap\mathcal{A}_j^{\mathrm{ar}}=\emptyset$ and $\mathcal{V}_i\cap\mathcal{V}_j=\emptyset$ hold for $i,j\in\{1,\dots,\nu\}$ with $i\neq j$ and we additionally have $\mathcal{A}^{\mathrm{ar}}=\dot{\bigcup}_{i=1}^{\nu}\mathcal{A}_i^{\mathrm{ar}}$.

The node set $\mathcal{V}_i$ can be further partitioned into fence nodes $\mathcal{V}_i^{\mathrm{fn}}$ and artificial nodes $\mathcal{V}_i^{\mathrm{ar}}$, i.e., $\mathcal{V}_i=\mathcal{V}_i^{\mathrm{fn}}\cup\mathcal{V}_i^{\mathrm{ar}}$. A node $v\in\mathcal{V}_i$ is called a fence node if it is connected to at least one pipeline outside the network station, i.e., if $\delta(v)\cap\mathcal{A}^{\mathrm{pi}}\neq\emptyset$. Hence, we note that a fence node is also part of the connecting network. Otherwise, if $\delta(v)\subseteq\mathcal{A}_i^{\mathrm{ar}}$, we call v an artificial node.

Furthermore, $\mathcal{F}_i\subseteq\mathcal{P}(\mathcal{V}_i^{\mathrm{fn}})\times\mathcal{P}(\mathcal{V}_i^{\mathrm{fn}})$ represents the set of so-called flow directions of network station G_i, where $\mathcal{P}$ denotes the powerset operator. A flow direction $f=(f^+,f^-)\in\mathcal{F}_i$ consists of its entry fence nodes $f^+\subseteq\mathcal{V}_i^{\mathrm{fn}}$ and its exit fence nodes $f^-\subseteq\mathcal{V}_i^{\mathrm{fn}}$ and it holds that $f^+\cap f^-=\emptyset$.

Finally, we are given a set $\mathcal{S}_i\subseteq\mathcal{P}(\mathcal{F}_i)\times\mathcal{P}(\mathcal{A}_i^{\mathrm{ar}})\times\mathcal{P}(\mathcal{A}_i^{\mathrm{ar}})$ of so-called simple states for each network station G_i. A simple state $s=(s^f,s^{\mathrm{on}},s^{\mathrm{off}})\in\mathcal{S}_i$ is composed of the set of flow directions s^f that it supports as well as the set of its active artificial arcs s^{on} and its inactive artificial arcs s^{off}. We assume that $s^{\mathrm{on}}\cup s^{\mathrm{off}}=\emptyset$, and we additionally call $\mathcal{A}_i^{\mathrm{ar}}\setminus(s^{\mathrm{on}}\cup s^{\mathrm{off}})$ the set of optional arcs.

Network Station Pressures

To differentiate between the role of the nodes in the connecting network and in the network stations, what is necessary to later on formulate flow direction related constraints in Subsection 3.5.4, we introduce a second class of pressure variables. For each $v \in \bigcup_{i \in I} \mathcal{V}_i$, i.e., every node contained in a network station, we introduce a pressure variable $\phi_{v,t} \in [\underline{p}_{v,t}, \bar{p}_{v,t}] \subseteq \mathbb{R}_{\geq 0}$ for each $t \in \mathcal{T}$. Here, $\underline{p}_{v,t}$ and $\bar{p}_{v,t}$ denote again technical pressure bounds. Moreover, we are given nonnegative initial pressure values $\phi_{v,0} \in \mathbb{R}_{\geq 0}$, too.

Thus, while for the artificial nodes $\mathcal{V}^{\text{ar}}$ we only have pressure variables $\phi_{v,t}$, for each fence node $v \in \mathcal{V}^{\text{fn}}$ we additionally have pressure variables $p_{v,t}$, too. In this context, we also note that $\phi_{v,0} = p_{v,0}$ holds for all fence nodes $v \in \mathcal{V}^{\text{fn}}$. The interplay of the two classes of pressure variables is discussed in Subsection 3.5.4.

Network Station Control

In each time step $t \in \mathcal{T}_0 := \{0, \dots, k\}$, three types of control decisions have to be taken for each network station G_i. These decisions impact each other and, in the following, we introduce the variables and constraints, modeling them and their interplay.

First, exactly one flow direction $f \in \mathcal{F}_i$ must be chosen for each G_i. Based on this, exactly one simple state $s \in \mathcal{S}_i$ which supports this flow direction has to be selected, i.e., $f \in s^f$ must hold. Next, given the simple state, all of its arcs in s^{on} must be active, while the inactive arcs s^{off} cannot be used. For the optional arcs $a \in \mathcal{A}_i^{\text{ar}} \setminus (s^{\text{on}} \cup s^{\text{off}})$ we can independently choose whether they are active or not.

Hence, for each time step $t \in \mathcal{T}_0$ we introduce binary variables $x_{f,t} \in \{0,1\}$ for each flow direction $f \in \mathcal{F}_i$, $x_{s,t} \in \{0,1\}$ for each simple state $s \in \mathcal{S}_i$, and $x_{a,t} \in \{0,1\}$ for each artificial arc $a \in \mathcal{A}_i^{\text{ar}}$ indicating whether the corresponding entity is selected (or active) or not. Additionally, for each network station G_i we add the following constraints

$$\sum_{f \in \mathcal{F}_i} x_{f,t} = 1 \qquad \forall t \in \mathcal{T}_0 \tag{3.6}$$

$$\sum_{s \in \mathcal{S}_i} x_{s,t} = 1 \qquad \forall t \in \mathcal{T}_0 \tag{3.7}$$

$$\sum_{f \in s^f} x_{f,t} \geq x_{s,t} \qquad \forall s \in \mathcal{S}_i,\ \forall t \in \mathcal{T} \tag{3.8}$$

$$x_{s,t} \leq x_{a,t} \qquad \forall s \in \mathcal{S}_i,\ \forall a \in s^{\text{on}},\ \forall t \in \mathcal{T}_0 \tag{3.9}$$

$$1 - x_{s,t} \geq x_{a,t} \qquad \forall s \in \mathcal{S}_i,\ \forall a \in s^{\text{off}},\ \forall t \in \mathcal{T}_0. \tag{3.10}$$

While constraints (3.6) and (3.7) ensure that exactly one flow direction and one simple state are chosen for each time step $t \in \mathcal{T}_0$, (3.8) guarantees that the chosen simple state supports the chosen flow direction. Additionally, constraints (3.9) and (3.10) make sure that the artificial arcs corresponding to the selected simple state

are active or not, respectively. No condition is imposed on the optional arcs. Note that important constraints related to the chosen flow direction are discussed in Subsection 3.5.4.

To penalize changes w.r.t. simple states or artificial arcs in the objective function, which correspond to changes in the settings of original network elements and are therefore strongly connected to our considered stability measure for the network control, we introduce additional binary variables and constraints. For each station G_i and each time step $t \in \mathcal{T}$ we add $\delta_{s,t} \in \{0,1\}$ for each $s \in \mathcal{S}_i$, $\delta^{\text{on}}_{a,t}, \delta^{\text{off}}_{a,t} \in \{0,1\}$ for each $a \in \mathcal{A}^{\text{ar}}_i$, and constraints

$$x_{s,t-1} - x_{s,t} + \delta_{s,t} \geq 0 \quad \forall s \in \mathcal{S}_i, \forall t \in \mathcal{T} \tag{3.11}$$

$$x_{a,t-1} - x_{a,t} + \delta^{\text{on}}_{a,t} - \delta^{\text{off}}_{a,t} = 0 \quad \forall a \in \mathcal{A}^{\text{ar}}_i, \forall t \in \mathcal{T}. \tag{3.12}$$

While $\delta_{s,t}$ and $\delta^{\text{on}}_{a,t}$ indicate whether a simple state or artificial arc has been switched on in time step t or not, $\delta^{\text{off}}_{a,t}$ shows whether an artificial arc has been switched off or not. For the simple states we do not need such variables, since we know that exactly one of them is active in each time step. All variables $\delta_{s,t}$ are associated with an individual cost parameter $w^s \in \mathbb{R}_{\geq 0}$, while variables $\delta^{\text{on}}_{a,t}$ as well as $\delta^{\text{off}}_{a,t}$ are assigned a cost parameters $w^a \in \mathbb{R}_{\geq 0}$.

Note that we refrain from penalizing changes w.r.t. flow directions. For example, if a network station is in a bypass state, e.g., only shortcuts are active, the gas may slosh back and forth between fence nodes. Thus, flow directions changes do not suggest an unstable behavior but are a common phenomenon.

Artificial Arcs

Next, we explain how the artificial arcs and their capabilities regarding the control of the gas flow through network stations are modeled. Recall that the set of artificial arcs can be partitioned into four disjoint subsets $\mathcal{A}^{\text{ar}} = \mathcal{A}^{\text{sc}} \cup \mathcal{A}^{\text{rg}} \cup \mathcal{A}^{\text{co}} \cup \mathcal{A}^{\text{cb}}$. $\mathcal{A}^{\text{sc}}$ denotes the set of shortcuts, $\mathcal{A}^{\text{rg}}$ the set of regulating arcs, $\mathcal{A}^{\text{co}}$ the set of compressor arcs, and $\mathcal{A}^{\text{cb}}$ the set of combined arcs. The sets $\mathcal{A}^{\text{sc}}_i \subseteq \mathcal{A}^{\text{sc}}$, $\mathcal{A}^{\text{rg}}_i \subseteq \mathcal{A}^{\text{rg}}$, $\mathcal{A}^{\text{co}}_i \subseteq \mathcal{A}^{\text{co}}$, and $\mathcal{A}^{\text{cb}}_i \subseteq \mathcal{A}^{\text{cb}}$ describe the corresponding entities contained in network station G_i. For each mono-directed arc $a \in \mathcal{A}^{\text{ar}}$ and each time step $t \in \mathcal{T}$ we introduce a variable $q_{a,t} \in [0, \bar{q}_{a,t}]$, which denotes the mass flow in forward direction w.r.t. the topological orientation. This flow is bounded by a given maximum mass flow parameter $\bar{q}_{a,t}$.

Bi-Directed Arcs

In contrast to mono-directed arcs, mass flow and pressure modifications according to the corresponding artificial arc type are possible in both directions on bi-directed arcs. Therefore, in our model, we choose a direction for each of these arcs and each point in time as follows: First, we replace each $a = (\ell, r) \in \mathcal{A}^{\text{bi}}$ by two anti-parallel mono-directed arcs $\overrightarrow{a} = (\ell, r)$ and $\overleftarrow{a} = (r, \ell)$ of the same type. Next, by introducing

binary variables $x_{\overrightarrow{a},t}, x_{\overleftarrow{a},t} \in \{0,1\}$ and constraints

$$x_{\overrightarrow{a},t} + x_{\overleftarrow{a},t} = x_{a,t} \quad \forall a \in \mathcal{A}^{\mathrm{bi}}, \forall t \in \mathcal{T}, \tag{3.13}$$

we determine the direction of use for each time step. Afterward, the models for the corresponding mono-directed arcs, which are explained in the following, apply.

Shortcuts

All shortcuts are bi-directed, i.e., we have $\mathcal{A}^{\mathrm{sc}} \subseteq \mathcal{A}^{\mathrm{bi}}$. They can conceptually be seen as the equivalent of valves inside a network station and can connect or disconnect parts of the network. In particular, after choosing a direction in (3.13), we add constraints

$$\phi_{\ell,t} - \phi_{r,t} \leq (1 - x_{a,t})(\bar{p}_{\ell,t} - \underline{p}_{r,t}) \quad \forall t \in \mathcal{T} \tag{3.14}$$
$$\phi_{\ell,t} - \phi_{r,t} \geq (1 - x_{a,t})(\underline{p}_{\ell,t} - \bar{p}_{r,t}) \quad \forall t \in \mathcal{T} \tag{3.15}$$
$$q_{a,t} \leq \bar{q}_{a,t}\, x_{a,t} \quad \forall t \in \mathcal{T} \tag{3.16}$$

for each (here conceptually mono-directed) shortcut $a = (\ell, r) \in \mathcal{A}^{\mathrm{sc}}$. If a shortcut is active at $t \in \mathcal{T}$, i.e., if $x_{a,t} = 1$, the pressures at ℓ and r are equal and mass flow in forward direction up to $\bar{q}_{a,t}$ is possible. If it is not active, the pressure values are decoupled, i.e., they are independent of each other, and there is no flow.

Regulating Arcs

Regulating arcs can be seen as the equivalent of regulators inside a network station. They are used to decrease the gas pressure in the direction of the flow. This ability is needed if, for example, gas enters a network part, which is not suited for higher pressures. Thus, for a regulating arc $a = (\ell, r) \in \mathcal{A}^{\mathrm{rg}}$, we introduce the following constraints

$$\phi_{\ell,t} - \phi_{r,t} \geq (1 - x_{a,t})(\underline{p}_{\ell,t} - \bar{p}_{r,t}) \quad \forall t \in \mathcal{T} \tag{3.17}$$
$$q_{a,t} \leq \bar{q}_{a,t}\, x_{a,t} \quad \forall t \in \mathcal{T}. \tag{3.18}$$

If a regulating arc is active at $t \in \mathcal{T}$, i.e., if $x_{a,t} = 1$, the pressure at ℓ has to be greater than or equal to the pressure at r and mass flow in forward direction up to $\bar{q}_{a,t}$ is possible. Otherwise, the pressures are decoupled and there is no mass flow.

Compressor Arcs

The compressor arcs $\mathcal{A}^{\mathrm{co}}$ are key elements when controlling gas transport networks. They can compress the gas and increase the pressure in the direction of the flow, which makes up for pressure loss due to friction in the pipelines or height differences.

In our model, one can conceptually think of one (big) compressor unit being installed on each arc $a \in \mathcal{A}^{\mathrm{co}}_i$ of each network station G_i. The maximum power

available for compression $\tilde{\pi}_{a,t} \in \mathbb{R}_{\geq 0}$, the maximum amount of mass flow that can pass through $\tilde{q}_{a,t} \in \mathbb{R}_{\geq 0}$, and the maximum compression ratio $\tilde{r}_{a,t} \in [1, \infty)$ are dynamically determined in each time step. Therefore, we consider approximations of real-world turbo compressor units for each network station, called machines in the following, which can be assigned to the arcs. Afterward, linear combinations of the corresponding values yield the parameters described above.

In particular, each network station G_i features a set of machines $\mathcal{M}_i$, and each machine $m \in \mathcal{M}_i$ possesses an associated maximum power $P_{m,t} \in \mathbb{R}_{\geq 0}$, a maximum mass flow $Q_{m,t} \in \mathbb{R}_{\geq 0}$, and a maximum compression ratio $R_{m,t} > 1$ for each time step $t \in \mathcal{T}$. Further, for each compressor arc $a \in \mathcal{A}_i^{\mathrm{co}}$, there exists a subset of machines $\mathcal{M}_i^a \subseteq \mathcal{M}_i$ that can potentially be assigned to it, and a maximum number of assignable machines $M_a^{\max}$. Since each machine can be assigned to at most one compressor arc in each time step $t \in \mathcal{T}$, we introduce binary variables $y_{m,a,t} \in \{0, 1\}$ indicating whether $m \in \mathcal{M}_i$ is assigned to $a \in \mathcal{A}_i^{\mathrm{co}}$ or not, and add constraints

$$\sum_{a \in \mathcal{A}_i^{\mathrm{co}} : m \in \mathcal{M}_i^a} y_{m,a,t} \leq 1 \qquad \forall m \in \mathcal{M}_i,\, \forall t \in \mathcal{T} \tag{3.19}$$

$$\sum_{m \in \mathcal{M}_i^a} y_{m,a,t} \leq M_a^{\max} x_{a,t} \quad \forall a \in \mathcal{A}_i^{\mathrm{co}},\, \forall t \in \mathcal{T}. \tag{3.20}$$

Recall that the compressor units can usually be operated individually, in parallel, sequentially, or in a parallel-sequential setting in the real world. This is achieved by the opening and closing of valves in the surrounding piping. By setting them up in parallel, a larger amount of mass flow can be compressed, while a higher compression ratio can be achieved sequentially. In our model, we refrain from choosing a setup for the machines and overestimate the capabilities of our compressor arcs in the sense that we assume that the maximum amount of flow (parallel setting) and the highest compression ratio (sequential setting) are available at the same time. Therefore, we add the following constraints

$$\sum_{m \in \mathcal{M}_i^a} P_{j,t}\, y_{m,a,t} = \tilde{\pi}_{a,t} \quad \forall a \in \mathcal{A}_i^{\mathrm{co}},\, \forall t \in \mathcal{T} \tag{3.21}$$

$$\sum_{m \in \mathcal{M}_i^a} Q_{j,t}\, y_{m,a,t} = \tilde{q}_{a,t} \quad \forall a \in \mathcal{A}_i^{\mathrm{co}},\, \forall t \in \mathcal{T} \tag{3.22}$$

$$1 + \sum_{m \in \mathcal{M}_i^a} (R_{j,t} - 1)\, y_{m,a,t} = \tilde{r}_{a,t} \quad \forall a \in \mathcal{A}_i^{\mathrm{co}},\, \forall t \in \mathcal{T}. \tag{3.23}$$

The first constraint (3.21) determines the power available on arc $a \in \mathcal{A}^{\mathrm{co}}$ by adding up the maximum powers of the assigned machines. Analogously, the second constraint (3.22) determines the maximum amount of flow that can pass through. On the other hand, the third constraint (3.23) is a linear approximation of the maximum compression ratio, which we use here to avoid the introduction of a nonlinear constraint.

The connection between the pressure ratio, the amount of mass flow passing through, and the power necessary to realize it is given by the nonlinear power equation for turbo compressor machines, see Hennings et al. [78] for example.

$$\tilde{\pi}_{a,t} \geq \pi_{a,t} = \frac{q_{a,t}}{\eta_{\text{ad}}} R_s T z_\ell \frac{\kappa}{\kappa - 1} \left[\left(\frac{\phi_{r,t}}{\phi_{\ell,t}} \right)^{\frac{\kappa-1}{\kappa}} - 1 \right] \tag{P}$$

Here, $\pi_{a,t} \in \mathbb{R}_{\geq 0}$ is the variable representing the necessary power when a mass flow of $q_{a,t}$ with initial pressure $\phi_{\ell,t}$ shall be compressed up to $\phi_{r,t}$. Further, η_{ad} is the adiabatic efficiency of the compression, which we assume to be constant for all compressor units. Moreover, we use $\kappa = 1.296$ for natural gas, see Fügenschuh et al. [50], and $\kappa = 1.5$ for hydrogen, see Stolten and Emonts [155], as isentropic exponents.

To once more avoid the introduction of a nonlinear constraint, we determine a linear approximation of (P) as follows: For each compressor arc $a \in \mathcal{A}^{\text{co}}$ and each $t \in \mathcal{T}$, we sample N points $(\phi_{\ell,t}, \phi_{r,t}, \pi_{a,t}) \in [\underline{p}_{\ell,t}, \bar{p}_{\ell,t}] \times [\underline{p}_{r,t}, \bar{p}_{r,t}] \times [\frac{\pi^{\max}_{a,t}}{4}, \pi^{\max}_{a,t}]$, where $\pi^{\max}_{a,t}$ is the maximum possible power for a at t derived from (3.19) and (3.20), such that $\phi_{\ell,t} \leq \phi_{r,t}$ and determine the corresponding mass flow $q_{a,t}$ using the original power equation. To the resulting set of 4-tuples we apply an ordinary least-squares method and determine coefficients $(\alpha_0, \alpha_1, \alpha_2, \alpha_3)$ for a linear approximation, which gives rise to constraints

$$\alpha_0 + \alpha_1 \phi_{\ell,t} + \alpha_2 \phi_{r,t} + q_{a,t} \leq \alpha_3 \pi_{a,t} + (1 - x_{a,t})(\alpha_0 + \alpha_1 \underline{p}_{\ell,t} + \alpha_2 \bar{p}_{r,t}), \tag{3.24}$$

$$\alpha_0 + \alpha_1 \phi_{\ell,t} + \alpha_2 \phi_{r,t} + q_{a,t} \geq \alpha_3 \pi_{a,t} + (1 - x_{a,t})(\alpha_0 + \alpha_1 \bar{p}_{\ell,t} + \alpha_2 \underline{p}_{r,t}). \tag{3.25}$$

We assume that $\alpha_1 \in \mathbb{R}_{\leq 0}$ and $\alpha_2 \in \mathbb{R}_{\geq 0}$ (otherwise we use the corresponding other bound for the coefficients of $x_{a,t}$ on the right-hand sides). If the compressor arc is active, it has to respect this linear approximation.

Next, we add the following set of constraints

$$\pi_{a,t} \leq \tilde{\pi}_{a,t} \qquad \forall a \in \mathcal{A}^{\text{co}}_i, \forall t \in \mathcal{T} \tag{3.26}$$

$$q_{a,t} \leq \tilde{q}_{a,t} \qquad \forall a \in \mathcal{A}^{\text{co}}_i, \forall t \in \mathcal{T} \tag{3.27}$$

$$\phi_{\ell,0} \tilde{r}_{a,t} - \phi_{r,t} \geq (1 - x_{a,t})(\phi_{\ell,0} - \bar{p}_{r,t}) \qquad \forall a \in \mathcal{A}^{\text{co}}_i, \forall t \in \mathcal{T}. \tag{3.28}$$

The first two (3.26) and (3.27) ensure that the mass flow and the power used for compression do not violate the upper bounds given by the machine assignments. Moreover, the outgoing pressure is bounded by the product of the initial ingoing pressure at $t = 0$ and the current maximum compression ratio (3.28) if the arc is active. Using the variable $\phi_{\ell,t}$ here instead of $\phi_{\ell,0}$ would yield a nonlinear constraint.

Besides constraints (3.19)–(3.28), for each compressor arc $a = (\ell, r) \in \mathcal{A}^{\text{co}}$ and

each time step $t \in \mathcal{T}$ we also add constraints

$$\phi_{\ell,t} - \phi_{r,t} \leq (1 - x_{a,t})(\bar{p}_{\ell,t} - \underline{p}_{r,t}) \qquad \forall t \in \mathcal{T} \tag{3.29}$$

$$r_{a,t}^{\max}\phi_{\ell,t} - \phi_{r,t} \geq (1 - x_{a,t})(r_{a,t}^{\max}\underline{p}_{\ell,t} - \bar{p}_{r,t}) \quad \forall t \in \mathcal{T}. \tag{3.30}$$

If the arc is active at some point in time $t \in \mathcal{T}$, i.e., $x_{a,t} = 1$, the pressure at ℓ has to be smaller than or equal to the pressure at r. Further, we bound $\phi_{r,t}$ by $r_{a,t}^{\max}\phi_{\ell,t}$. Here, $r_{a,t}^{\max}$ is the maximum possible compression ratio of a at time t, which can be derived from constraints (3.20) and (3.23). If it is not active, the pressure values are decoupled and there is no mass flow due to constraints (3.20),(3.22), and (3.27).

Combined Arcs

Finally, a combined arc $a = (\ell, r) \in \mathcal{A}^{\text{cb}}$ can either be used as regulating arc or as compressor arc. Thus, in our model we replace each combined arc by two corresponding parallel arcs, i.e., a regulating arc $a^{\text{rg}} = (\ell, r) \in \mathcal{A}^{\text{rg}}$ and a compressor arc $a^{\text{cp}} = (\ell, r) \in \mathcal{A}^{\text{co}}$. Using binary variables $x_{a^{\text{rg}},t}, x_{a^{\text{cp}},t} \in \{0, 1\}$ and constraints

$$x_{a^{\text{rg}},t} + x_{a^{\text{cp}},t} = x_{a,t} \quad \forall a \in \mathcal{A}^{\text{cb}}, \forall t \in \mathcal{T}, \tag{3.31}$$

we decide which mode of the combined arc is active in each time step. The corresponding model is then applied accordingly.

Flow Direction Related Constraints

Next, activating a flow direction imposes certain conditions on the mass flows into and out of a network station G_i. Most importantly, for a flow direction $f = (f^+, f^-) \in \mathcal{F}_i$, no outflow is allowed at its entry fence nodes and no inflow is allowed at its exit fence nodes. It is, however, allowed that there is no flow at all, which is the condition that has to hold for all remaining fence nodes $v \in \mathcal{V}_i^{\text{fn}} \setminus (f^+ \cup f^-)$.

Inflow and Outflow Constraints

For each fence node $v \in \mathcal{V}_i^{\text{fn}}$ and each point in time $t \in \mathcal{T}_0$, we introduce two continuous variables $q_{v,t}^{\text{in}}, q_{v,t}^{\text{out}} \in \mathbb{R}_{\geq 0}$. Together with the following constraint, they account for the total inflow or outflow from the connecting network

$$\sum_{a=(\ell,v)\in\mathcal{A}^{\text{ar}}} q_{a,t} - \sum_{a=(v,r)\in\mathcal{A}^{\text{ar}}} q_{a,t} = q_{v,t}^{\text{out}} - q_{v,t}^{\text{in}} \quad \forall v \in \mathcal{V}^{\text{fn}}, \forall t \in \mathcal{T}_0. \tag{3.32}$$

Note that one could alternatively sum up the mass flow values at the incident pipelines on the left-hand side and switch the signs of the variables on the right-hand side of the equation. This is because flow conservation, which we introduce as constraints (3.38) in Subsection 3.5.4, holds for $\mathcal{V}_i^{\text{fn}} \subseteq \mathcal{V}^0$. Finally, for each flow

direction $f = (f^+, f^-) \in \mathcal{F}_i$, we introduce the following constraints

$$q^{\text{in}}_{v,t} \leq \overline{q}^{\text{in}}_{v,t}\,(1 - x_{f,t}) \quad \forall i \in I,\, \forall f \in \mathcal{F}_i,\, \forall v \in \mathcal{V}^{\text{fn}}_i \setminus f^+,\, \forall t \in \mathcal{T} \tag{3.33}$$

$$q^{\text{out}}_{v,t} \leq \overline{q}^{\text{out}}_{v,t}\,(1 - x_{f,t}) \quad \forall i \in I,\, \forall f \in \mathcal{F}_i,\, \forall v \in \mathcal{V}^{\text{fn}}_i \setminus f^-,\, \forall t \in \mathcal{T}. \tag{3.34}$$

Here, $\overline{q}^{\text{in}}_{v,t}$ and $\overline{q}^{\text{out}}_{v,t}$ are upper and lower bounds on the maximum possible inflow and outflow, respectively. Both can be derived from the above-mentioned alternative constraint. If a flow direction is active, $q^{\text{in}}_{v,t}$ can only be nonzero for entry fence groups and $q^{\text{out}}_{v,t}$ only for exit fence groups.

Exit Fence Node Pressure Bounds

For some fence nodes $v \in \mathcal{V}^{\text{fn}}_i$, there exists an additional upper pressure bound $\bar{\phi}^{\text{exit}}_v$. It is tighter than the corresponding technical bound and must be respected if a flow direction $f = (f^+, f^-) \in \mathcal{F}_i$ is active for which v is an exit fence node, i.e., $v \in f^-$. This can be modeled with constraints

$$\phi_{v,t} \leq \bar{p}_{v,t} + x_{f,t}\,(\bar{\phi}^{\text{exit}}_v - \bar{p}_{v,t}) \quad \forall i \in I,\, \forall f \in \mathcal{F}_i, \forall v \in f^-,\, \forall t \in \mathcal{T}. \tag{3.35}$$

3.5.4. Linking the Connecting Network and the Network Stations

The two models for the connecting network and the network stations are linked through the following two classes of constraints.

Coupling Fence Node Pressures

Depending on the flow direction, the two classes of pressure variables for fence nodes must be equal or are decoupled. In particular, $p_{v,t}$ and $\phi_{v,t}$ have to be equal for $v \in \mathcal{V}^{\text{fn}}$ if it is either an entry or an exit of the flow direction that is active at time step $t \in \mathcal{T}$. Otherwise, the pressures are decoupled. This is modeled via constraints

$$\phi_{v,t} - p_{v,t} \leq (1 - x_{f,t})(\bar{p}_{v,t} - \underline{p}_{v,t}) \quad \forall f \in \mathcal{F}_i,\, \forall v \in \mathcal{V}^{\text{fn}}_i \setminus (f^- \cup f^+),\, \forall t \in \mathcal{T} \tag{3.36}$$

$$\phi_{v,t} - p_{v,t} \geq (1 - x_{f,t})(\underline{p}_{v,t} - \bar{p}_{v,t}) \quad \forall f \in \mathcal{F}_i,\, \forall v \in \mathcal{V}^{\text{fn}}_i \setminus (f^- \cup f^+),\, \forall t \in \mathcal{T}. \tag{3.37}$$

This behavior can be interpreted as having a valve between the fence nodes and the pipelines incident to them. If a fence node is an entry or exit of the active flow direction, mass flow from or into the incident pipelines is possible, and the pressures are equal. Otherwise, no mass flow is possible, and the pressures are decoupled.

Mass Flow Conservation

Finally, for all nodes $v \in \mathcal{V}$ we introduce mass flow conservation constraints. For each inner node $v \in \mathcal{V}^0$ and each time step $t \in \mathcal{T}$, the amount of flow entering v has to leave it, and for each boundary node $v \in \mathcal{V}^{\text{b}}$, the supply or demand must be met.

Thus, we derive

$$\sum_{a=(v,r)\in\mathcal{A}^{\mathrm{pi}}} q_{v,a,t} + \sum_{a=(v,r)\in\mathcal{A}^{\mathrm{ar}}} q_{a,t} - \sum_{a=(\ell,v)\in\mathcal{A}^{\mathrm{pi}}} q_{v,a,t} - \sum_{a=(\ell,v)\in\mathcal{A}^{\mathrm{ar}}} q_{a,t} = 0 \qquad \forall v \in \mathcal{V}^0,\, \forall t \in \mathcal{T} \tag{3.38}$$

$$\sum_{a=(v,r)\in\mathcal{A}^{\mathrm{pi}}} q_{v,a,t} - \sum_{a=(\ell,v)\in\mathcal{A}^{\mathrm{pi}}} q_{v,a,t} = d_{v,t} \qquad \forall v \in \mathcal{V}^{\mathrm{b}},\, \forall t \in \mathcal{T}. \tag{3.39}$$

Hence, the mass flow conservation constraints for the fence nodes $v \in \mathcal{V}^{\mathrm{fn}} \subseteq \mathcal{V}^0$ connect the two models for the connecting network and the network stations.

3.5.5. Objective Functions and Complete Model

To conclude the definition of our tri-level MIP model, we finally turn to its three objective functions. Recall the rationale behind our formulation as explained in Section 3.4. In the real world, dispatchers try to control the network using technical measures only, i.e., changing the settings of the remotely controllable elements to satisfy the supplies and demands. If this does not work, they have several non-technical measures at hand. The most common and standardized ones are those changing supplies and demands by applying contractual options like the interruption of customers or buying or selling gas, e.g., using so-called balancing energy. If changing the supplies and demands does not help, the last option is to ask other transport system operators for changes in future entry pressures. In practice, this is done by phone calls, and it can therefore be seen as a last possible non-standardized option. Therefore, we state the complete tri-level MIP as

$$\min_{\sigma^p} \sum_{t\in\mathcal{T}} \sum_{v\in\mathcal{V}^{\mathrm{b}}} (\sigma^{p+}_{v,t} + \sigma^{p-}_{v,t}) \tag{3.40}$$

$$\min_{\sigma^d} \sum_{t\in\mathcal{T}} \sum_{v\in\mathcal{V}^{\mathrm{b}}} (\sigma^{d+}_{v,t} + \sigma^{d-}_{v,t}) \tag{3.41}$$

$$\min_{\dots} \sum_{t\in\mathcal{T}} \Big(\sum_{s\in\mathcal{S}} w^s \delta_{s,t} + \sum_{a\in\mathcal{A}^{\mathrm{ar}}} w^a (\delta^{\mathrm{on}}_{a,t} + \delta^{\mathrm{off}}_{a,t}) \Big) \tag{3.42}$$

$$\text{s.t. } (3.1) - (3.39)$$

The first level controls the inflow pressure slack variables $\sigma^{p+}_{v,t}$ and $\sigma^{p-}_{v,t}$. The second level controls the boundary flow slack variables $\sigma^{d+}_{v,t}$ and $\sigma^{d-}_{v,t}$. The goal of both is to minimize the corresponding sums, i.e., the sum of absolute deviations, see (3.40) and (3.41), respectively. The third level, which is responsible for the technical control of the network while the upper two ensure feasibility, controls all other variables. Its goal is to determine a control with maximum stability for the network. Accordingly, it minimizes the weighted sum of changes regarding the simple states and artificial arcs, and reflects the stability indicator proposed by the practitioners, see (3.42).

3.6. An Algorithm for the Tri-Level MIP

In this section, we present an algorithm to solve the tri-level MIP introduced in the previous section. It is based on solving a series of closely-related single-level MIPs. Therefore, we first define MIP L_3, which consists of third level's objective function (3.42), constraint set (3.1)–(3.39), and both classes of slack variables being fixed to zero. Second, the MIP given by the second level's objective (3.41), constraints (3.1)–(3.39), and all inflow pressure slack variables being fixed to zero, we denote by L_2. Third, the first level's objective (3.40) combined with (3.1)–(3.39) represents MIP formulation L_1. In the context of hierarchical optimization, L_1 is called the high point relaxation. The procedure described in Algorithm 1 can then be used to solve the tri-level MIP.

Algorithm 1: An algorithm for the tri-level MIP model

Input : The tri-level MIP model
Output: An optimal solution for it or `INFEASIBLE`

```
1  if L3 is infeasible then
2      if L2 is infeasible then
3          if L1 is infeasible then
4              return INFEASIBLE
5          else
6              SOL1 ← Optimal solution for L1
7              L̃2 ← L2 with inflow pressure slack variables fixed to SOL1
8              SÕL2 ← Optimal solution for L̃2
9              L̃3 ← L3 with both classes of slack variables fixed to SÕL2
10             return Optimal solution for L̃3
11     else
12         SOL2 ← Optimal solution for L2
13         L̂3 ← L3 with boundary flow slack variables fixed to SOL2
14         return Optimal solution for L̂3
15 else
16     return Optimal solution for L3
```

If there exists a feasible solution with no slacks, i.e., for L_3, an optimal solution for it is an optimal solution for the tri-level MIP, and it is returned in line 16. Otherwise, if there exists a feasible solution for L_2, we subsequently solve $\hat{L}_3$, i.e., L_3 with all slack variables being fixed to an optimal solution of L_2. Doing this, we again determine an optimal solution for the hierarchical MIP formulation, see lines 12–14. Finally, if L_2 does not admit a feasible solution, we consider the high point relaxation L_1. If it is infeasible, the tri-level MIP itself is infeasible. Otherwise, we subsequently solve $\tilde{L}_2$ and $\tilde{L}_3$ and determine an optimal solution, see lines 6–10.

3.6.1. Rolling Horizon Heuristic with Backtracking

Determining feasible solutions for the single-level MIPs introduced above before starting a branch-and-bound algorithm to solve them is beneficial in two aspects. Besides showing that the problem is not infeasible, the solution can be used as incumbent and accelerate the solving process. This is the motivation for the introduction of our rolling horizon heuristic. The idea to use this concept here comes from the fact that rolling horizon has been successively applied to various time-dependent optimization problems in the past, e.g., disruption management in the railway industry, see Nielsen et al. [116], or several kinds of scheduling problems, see for example Addis et al. [3] and Samà et al. [144].

Our heuristic is stated in Algorithm 2. Here, MIP_k^j denotes the MIP model for the first k time steps with all binary variables corresponding to time steps $i \in \mathcal{T}_0$ with $i < j$ being fixed.

Algorithm 2: Rolling Horizon Heuristic with Backtracking

Input : MIP model $\text{L}_1, \text{L}_2, \tilde{\text{L}}_2, \text{L}_3, \tilde{\text{L}}_3$, or $\hat{\text{L}}_3$ from Algorithm 1
Output: A feasible solution or `INFEASIBLE`

1 $\text{MIP}_0^0 \leftarrow$ MIP model for time step 0
2 **if** *MIP_0^0 is infeasible* **then**
3 `INFEASIBLE`
4 $S_0 \leftarrow$ Optimal solution for MIP_0^0
5
6 **for** $k \leftarrow 1$ **to** n **do**
7 $\text{MIP}_k^0 \leftarrow$ MIP model for time steps $\{0, \dots, k\}$
8 $\text{MIP}_k^k \leftarrow$ Fix binary variables for time steps $i < k$ in MIP_k^0 to S_{k-1}
9 $j \leftarrow k$
10 **while** *MIP_k^j is infeasible* **do**
11 $j \leftarrow j - 1$
12 **if** $j = 0$ **then**
13 **return** `INFEASIBLE`
14 $\text{MIP}_k^j \leftarrow$ Fix binary variables for time steps $i < j$ in MIP_k^0 to S_{k-1}
15 $S_k \leftarrow$ Optimal solution for MIP_k^j
16 **return** S_n

Starting with time step $k = 0$, in each of the following iterations, we consider the single-level MIP model MIP_k^j with an additional time step and solve it. Here, we fix all binary variables to the solution values of the corresponding binary variables of an optimal solution of the previous iteration, i.e., only the binary variables of the newly added time step are not fixed, see lines 6–10 and 15.

Additionally, we apply backtracking within our heuristic, which is used whenever a MIP turns out to be infeasible, see lines 10–13. In that case, we iteratively release (unfix) the binary variables of preceding time steps until a feasible solution is found.

Afterward, we continue with the rolling horizon approach as usual. Note that if all binary variables are released, and the current MIP model is still infeasible, this shows that the original MIP is infeasible as MIP_k^1 is an inconsistent subsystem in the original model, see the PhD thesis of Pfetsch [131] for more details on this topic.

If we impose a time limit on the MIP solves within Algorithm 2 and run into it, the heuristic continues with the incumbent solution. If no feasible solution is found, we consider the corresponding MIP to be infeasible.

3.6.2. Solution Smoothing

Due to the nature of many LP-based branch-and-bound algorithms, a phenomenon that can occur is the non-smoothness of the obtained solutions. As an example, we can observe on the compressor arcs that in many cases, massive amounts of gas are compressed in a single time step, while there is no compression at all in all other time steps. The same behavior can be observed for the outgoing pressures of the compressor arcs as well. Of course, these solutions with considerable differences in the corresponding variable values for consecutive time steps are feasible w.r.t. the model. However, such a behavior is not desirable in practice. Instead, we would like to have constant mass flow and pressure at the fence nodes of the network stations during the whole time horizon, if this is possible. Considering the extent of absolute changes in the flow and pressure values at the fence nodes of a network stations for consecutive time steps as a measure for the smoothness of a solution, we derive the following LP formulation to determine the smoothest solution w.r.t. fixed binary decisions.

Given a solution S for the hierarchical MIP formulation, consider the single-level LP, which consists of the constraints and variables of the third level, but where we additionally fix all binary variables and slack variables to their corresponding solution values in S.

Furthermore, for each network station G_i, each fence node $v \in \mathcal{V}_i^{\text{fn}}$, and each time step $t \in \mathcal{T}$, we add four continuous variables $\delta_{v,t}^{p+}, \delta_{v,t}^{p-}, \delta_{v,t}^{q+}, \delta_{v,t}^{q-} \in \mathbb{R}_{\geq 0}$, two continuous variables $\overline{\delta}_v^p, \overline{\delta}_v^q \in \mathbb{R}_{\geq 0}$, and the following constraints

$$p_{v,t} - p_{v,t-1} = \delta_{v,t}^{p+} - \delta_{v,t}^{p-} \quad \forall v \in \mathcal{V}_i^{\text{fn}}, \forall t \in \mathcal{T} \tag{3.43}$$

$$q_{v,t}^{\text{out}} - q_{v,t}^{\text{in}} - q_{v,t-1}^{\text{out}} + q_{v,t-1}^{\text{in}} = \delta_{v,t}^{q+} - \delta_{v,t}^{q-} \quad \forall v \in \mathcal{V}_i^{\text{fn}}, \forall t \in \mathcal{T} \tag{3.44}$$

$$\delta_{v,t}^{p+} + \delta_{v,t}^{p-} \leq \overline{\delta}_v^p \quad \forall v \in \mathcal{V}_i^{\text{fn}}, \forall t \in \mathcal{T} \tag{3.45}$$

$$\delta_{v,t}^{q+} + \delta_{v,t}^{q-} \leq \overline{\delta}_v^q \quad \forall v \in \mathcal{V}_i^{\text{fn}}, \forall t \in \mathcal{T}. \tag{3.46}$$

Constraint (3.43) measures the difference between the pressure values at v from time step $t-1$ to time step t using variables $\delta_{v,t}^{p+}$ and $\delta_{v,t}^{p-}$. Analogously, the difference between inflow and outflow is measured by constraint (3.44) using $\delta_{v,t}^{q+}$ and $\delta_{v,t}^{q-}$. Moreover, the maximum absolute difference between any two consecutive time steps w.r.t. pressure as well as inflow and outflow at v is determined by constraints (3.45) and (3.46) using variables $\overline{\delta}_v^p$ and $\overline{\delta}_v^q$, respectively.

Further, for each network station G_i and each time step $t \in \mathcal{T}$, we add two continuous variables $\overline{\delta}^p_{i,t}, \overline{\delta}^q_{i,t} \in \mathbb{R}_{\geq 0}$, and introduce constraints

$$\delta^{p+}_{v,t} + \delta^{p-}_{v,t} \leq \overline{\delta}^p_{i,t} \quad \forall v \in \mathcal{V}^{\text{fn}}_i, \forall t \in \mathcal{T} \tag{3.47}$$

$$\delta^{q+}_{v,t} + \delta^{q-}_{v,t} \leq \overline{\delta}^q_{i,t} \quad \forall v \in \mathcal{V}^{\text{fn}}_i, \forall t \in \mathcal{T}. \tag{3.48}$$

Here, the maximum difference w.r.t. pressure as well as inflow and outflow between time step $t-1$ and time step t over all fence nodes of station G_i is determined using variables $\overline{\delta}^p_{i,t}$ and $\overline{\delta}^q_{i,t}$, respectively.

While the two variables $\overline{\delta}^p_v$ and $\overline{\delta}^q_v$ are associated with positive objective coefficients $w^{\text{sm-p}}_v, w^{\text{sm-q}}_v \in \mathbb{R}_{>0}$, the variables $\overline{\delta}^p_{i,t}$ and $\overline{\delta}^q_{i,t}$ are associated with positive objective coefficients $w^{\text{sm-p}}_i, w^{\text{sm-q}}_i \in \mathbb{R}_{>0}$, respectively. Note that these are the only variables with nonzero objective coefficients. Preliminary experiments showed that only penalizing the maximum differences per station can lead to a majority of the stations behaving very smoothly, while some show significant differences. On the other hand, by only penalizing the maximum differences, we could give away smoothness potential w.r.t. the single stations.

Finally, we denote the linear program described here by $\text{LP}_{\text{sm}}(S)$ in the following, i.e., the linear program to smooth S. Since it is based on a feasible solution and bounded by construction, we note that it admits an optimal solution.

3.7. An Iterative Velocity Adjustment Procedure

A possible drawback of the applied linear model for the transient gas flow through pipelines is the fixation of the absolute velocity in the friction term of the Momentum Equation (3.5), see the discussion on the pipeline model in Subsection 3.5.2. If the mass flows or pressures at the endnodes of a pipeline change significantly in future time steps, we may underestimate or overestimate the pressure loss, what can in turn lead to inaccurate control decisions. To prevent this, we introduce an iterative velocity adjustment procedure (IVAP) as postprocessing step in this section. Inspired by the successful application of sequential linear programming in the context of gas network control problems, see for example González-Rueda et al. [63], its goal is to derive solutions that are feasible for the tri-level MIP with constraints (3.5) replaced by the nonlinear constraints (M).

The main idea of the IVAP is the following: Given a solution S for the tri-level MIP model, consider the LP derived from the variables and constraints of the third level and fixing all binary variables to their corresponding solution values. Next, we update the friction terms of the linear model for the Momentum Equations by using the gas velocities at the endnodes of all pipelines derived from S instead of the velocities based on the initial state. The goal now is to determine a solution such that the pressure and flow variables in constraints (3.5) stay as close as possible to the corresponding solution values in S. If we can find a solution such that all these variable values are the same, we have determined a solution satisfying constraints (M),

as the gas velocity depends on the mass flow and pressure at the corresponding node only, see equation (Q). Therefore, to derive such a solution, we introduce additional variables and constraints to measure deviations, and penalize them through the LP's objective function. In the following, for all nodes $v \in \mathcal{V}^{\text{pi}}$, let us denote the solution values of the corresponding variables in S by ${}^{S}p_{v,t}$, ${}^{S}q_{a,\ell,t}$, and ${}^{S}q_{a,r,t}$.

First, for each $v \in \mathcal{V}^{\text{pi}}$, i.e., each node incident to a pipeline, and each time step $t \in \mathcal{T}$, we add two variables $\delta^{p+}_{v,t}, \delta^{p-}_{v,t} \in \mathbb{R}_{\geq 0}$. Furthermore, we add one additional variable $\overline{\delta}^{p} \in \mathbb{R}_{\geq 0}$ and constraints

$$p_{v,t} - {}^{S}p_{v,t} = \delta^{p+}_{v,t} - \delta^{p-}_{v,t} \quad \forall v \in \mathcal{V}^{\text{pi}},\, \forall t \in \mathcal{T} \tag{3.49}$$

$$\delta^{p+}_{v,t} + \delta^{p-}_{v,t} \leq \overline{\delta}^{p} \quad \forall v \in \mathcal{V}^{\text{pi}},\, \forall t \in \mathcal{T}. \tag{3.50}$$

Constraint (3.49) measures the deviation of the pressure value of node v and time step t from the corresponding solution value in S using variables $\delta^{p+}_{v,t}$ and $\delta^{p-}_{v,t}$. The maximum difference for any node and any time step is then determined by constraints (3.50), and it is equal to $\overline{\delta}^{p}$ since we assign a positive objective coefficient $\overline{w}^{\text{sm-p}} \in \mathbb{R}_{>0}$ to it. Moreover, we also introduce an objective coefficient $w^{\text{sm-p}} \in \mathbb{R}_{>0}$ for all variables $\delta^{p+}_{v,t}$ and $\delta^{p-}_{v,t}$.

Next, for each pipeline $a = (\ell, r) \in \mathcal{A}^{\text{pi}}$ and each time step $t \in \mathcal{T}$, we add four continuous variables $\delta^{q+}_{a,\ell,t}, \delta^{q-}_{a,\ell,t}, \delta^{q+}_{a,r,t}, \delta^{q-}_{a,r,t} \in \mathbb{R}_{\geq 0}$. Furthermore, we add one variable $\overline{\delta}^{q} \in \mathbb{R}_{\geq 0}$ and constraints

$$q_{a,\ell,t} - {}^{S}q_{a,\ell,t} = \delta^{q+}_{a,\ell,t} - \delta^{q-}_{a,\ell,t} \quad \forall a = (\ell, r) \in \mathcal{A}^{\text{pi}},\, \forall t \in \mathcal{T} \tag{3.51}$$

$$q_{a,r,t} - {}^{S}q_{a,r,t} = \delta^{q+}_{a,r,t} - \delta^{q-}_{a,r,t} \quad \forall a = (\ell, r) \in \mathcal{A}^{\text{pi}},\, \forall t \in \mathcal{T} \tag{3.52}$$

$$\delta^{q+}_{a,\ell,t} + \delta^{q-}_{a,\ell,t} \leq \overline{\delta}^{q} \quad \forall a = (\ell, r) \in \mathcal{A}^{\text{pi}},\, \forall t \in \mathcal{T} \tag{3.53}$$

$$\delta^{q+}_{a,r,t} + \delta^{q-}_{a,r,t} \leq \overline{\delta}^{q} \quad \forall a = (\ell, r) \in \mathcal{A}^{\text{pi}},\, \forall t \in \mathcal{T}. \tag{3.54}$$

Here, constraints (3.51) and (3.52) determine the deviation of the inflow or outflow at the endnodes ℓ and r of pipeline a from the corresponding solution values in S. The maximum difference for any pipe, any endnode, and any time step is determined using constraints (3.53) and (3.54), and it is equal to $\overline{\delta}^{q}$ because we assign a positive objective coefficient $\overline{w}^{\text{sm-q}} \in \mathbb{R}_{>0}$ to it. For variables $\delta^{q+}_{a,\ell,t}, \delta^{q-}_{a,\ell,t}, \delta^{q+}_{a,r,t}, \delta^{q-}_{a,r,t} \in \mathbb{R}_{\geq 0}$, we additionally introduce an objective coefficient $w^{\text{sm-q}} \in \mathbb{R}_{>0}$. Finally, zero is assigned as objective coefficient to all other variables. We denote the resulting linear program by $\text{LP}_{\text{base}}(S)$ in the following.

However, first experiments with $\text{LP}_{\text{base}}(S)$ showed that many slack variables become nonzero or attain solution values in higher magnitudes compared to S in the corresponding solutions, since they do not contribute to the objective function anymore. On the other hand, when fixing all slack variables to their corresponding solution values in S, the resulting LP formulations often become infeasible. These observations gave rise to a middle ground. For both classes of slack variables, we introduce a parameter $\gamma \in \mathbb{R}_{\geq 0}$ with $\gamma \geq 1$, and given the corresponding solution

Algorithm 3: Iterative Velocity Adjustment Procedure (IVAP)

Input : Solution S_0 for tri-level MIP as well as parameters ε, γ, μ, and Δ
Output: Feasible solution where (3.5) is replaced by (M) or `UNSUCCESSFUL`

$v_0 \leftarrow$ Gas velocities at $v \in \mathcal{V}^{\text{pi}}$ w.r.t. S_0
$i \leftarrow 1$
while $i \leq \Delta$ **do**
  $v_i^* \leftarrow \sum_{j=\max\{0,i-\mu\}}^{i-1} \frac{|v_j|}{\min\{i,\mu\}}$
  $\text{LP}_{\text{iv}} \leftarrow \text{LP}_{\text{base}}(S_{i-1})$ with $|v_i^*|$ used in (3.5) and $\text{MG}(S_0, \gamma)$
  if *LP_{iv} is infeasible* **then**
    return `UNSUCCESSFUL`
  $S_i \leftarrow$ Optimal solution for LP_{iv}
  $v_i \leftarrow$ Gas velocities at $v \in \mathcal{V}^{\text{pi}}$ w.r.t. S_i
  if $|||v_i| - |v_i^*|||_\infty \leq \varepsilon$ **then**
    return S_i
  $i \leftarrow i + 1$
return `UNSUCCESSFUL`

values ${}^S\sigma_{v,t}^{d+}$, ${}^S\sigma_{v,t}^{d-}$, ${}^S\sigma_{v,t}^{p+}$, and ${}^S\sigma_{v,t}^{p-}$, we add constraints

$$\sigma_{v,t}^{d+} \leq \gamma \cdot {}^S\sigma_{v,t}^{d+} \quad \forall v \in \mathcal{V}^{\text{b}}, \forall t \in \mathcal{T} \tag{3.55}$$
$$\sigma_{v,t}^{d-} \leq \gamma \cdot {}^S\sigma_{v,t}^{d-} \quad \forall v \in \mathcal{V}^{\text{b}}, \forall t \in \mathcal{T} \tag{3.56}$$
$$\sigma_{v,t}^{p+} \leq \gamma \cdot {}^S\sigma_{v,t}^{p+} \quad \forall v \in \mathcal{V}^{\text{b}}, \forall t \in \mathcal{T} \tag{3.57}$$
$$\sigma_{v,t}^{p-} \leq \gamma \cdot {}^S\sigma_{v,t}^{p-} \quad \forall v \in \mathcal{V}^{\text{b}}, \forall t \in \mathcal{T}. \tag{3.58}$$

This approach fixes the slack variables with solution value zero, while all other slack variables stay within a controllable range w.r.t. γ in the resulting LP. We denote this set of constraints (3.55)–(3.58) by $\text{MG}(S, \gamma)$ in the following.

The final, complete iterative velocity adjustment procedure (IVAP) is stated in Algorithm 3. Given a feasible solution S_0 for the initial tri-level MIP model, we determine the gas velocities for all $v \in \mathcal{V}^{\text{pi}}$ and all time steps. Next, we repeat the following procedure: In iteration i, we determine the average absolute gas velocity of the last $\min\{i, \mu\}$ solutions, i.e., v_i^*. Afterward, we obtain the linear program LP_{iv} as $\text{LP}_{\text{base}}(S_{i-1})$ using $|v_i^*|$ in constraints (3.5) together with constraints $\text{MG}(S_0, \gamma)$. If LP_{iv} is infeasible, the procedure is terminated and `UNSUCCESSFUL` is returned. Otherwise we retrieve the gas velocities v_i from an optimal solution S_i. If $|v_i|$ and $|v_i^*|$ differ by less than ε for all pipelines, nodes and time steps, S_i is returned as result. If this criterion is not satisfied within Δ iterations, the procedure is aborted and `UNSUCCESSFUL` is returned.

Algorithm 4: Algorithmic Framework

Input : Tri-level MIP formulation and parameters Δ, ε, γ, and μ
Output: Solution for tri-level MIP with constraints (M) or `UNSUCCESSFUL`

1 $i \leftarrow 0$
2 **while** $i \leq \Delta$ **do**
3 $\quad i \leftarrow i + 1$
4 $\quad$ **if** *the tri-level MIP is infeasible* **then**
5 $\quad\quad$ **return** `UNSUCCESSFUL`
6 $\quad S_i \leftarrow$ Optimal solution for the tri-level MIP
7 $\quad S_i^{\mathrm{sm}} \leftarrow$ Smoothening routine applied to S_i
8 $\quad S_i^{\mathrm{iv}} \leftarrow \mathrm{IVAP}(S_i^{\mathrm{sm}}, \varepsilon, \gamma, \mu)$
9 $\quad$ **if** $S_i^{iv} \neq$ *`UNSUCCESSFUL`* **then**
10 $\quad\quad$ **return** S_i^{iv}
11 $\quad v_i^{\mathrm{update}} \leftarrow$ Gas velocities at $v \in \mathcal{V}^{\mathrm{pi}}$ from last IVAP iteration
12 $\quad$ Update Momentum Equations (3.5) in the tri-level MIP with v_i^{update}
13 $\quad$ **if** *three solutions with variable values* ${}^{S_i}x_{s,t}$ *have been considered* **then**
14 $\quad\quad$ Add no-good-cut w.r.t. ${}^{S_i}x_{s,t}$
15 **return** `UNSUCCESSFUL`

3.8. Complete Algorithmic Approach

Our complete algorithmic approach for determining a stable transient control of gas transport networks is presented in Algorithm 4.

The following procedure is iteratively repeated: In iteration i and lines 4–8, we solve the tri-level MIP model with Algorithm 1 from Section 3.6. If the model is infeasible, `UNSUCCESSFUL` is returned. Otherwise, we apply the smoothing routine from Subsection 3.6.2 to the obtained solution S_i, and subsequently start the IVAP with the resulting solution S_i^{sm}, see Section 3.7.

If the IVAP terminates with a feasible solution, it is a feasible solution for the tri-level MIP with constraints (3.5) being replaced with (M) and Algorithm 4 terminates, see line 10. Otherwise, we retrieve the velocities v_i^{update} from the IVAP iteration executed last, and derive a new tri-level MIP formulation where constraints (3.5) are updated using the corresponding absolute values in the friction terms, see lines 11 and 12. Additionally, we add a so-called no-good-cut w.r.t. the simple state variables of solution S_i, which we denote by ${}^{S_i}x_{s,t}$, if we have obtained three solutions coinciding in these values. In particular, we add the constraint

$$\sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}_0} {}^{S_i}x_{s,t} \cdot x_{s,t} \leq n \cdot \nu - 1,$$

where $n = |\mathcal{T}_0|$ is the number of time steps and ν the number of network stations, see line 14. Moreover, the algorithm terminates with `UNSUCCESSFUL` if no solution

has been found within Δ iterations.

The reason for starting the IVAP with the smoothed solution S_i^{sm} instead of S_i is that the differences w.r.t. the gas velocities at the nodes for consecutive time steps are intuitively smaller. Furthermore, the solution produced by the IVAP may maintain some degree of the smoothness of the initial S_i^{sm}.

If the IVAP fails, we use the velocities from its last iteration in the friction terms of the Momentum Equations of the tri-level model in the next iteration, since they may be closer to the velocities of some feasible solution for the model with the nonlinear constraints (M). Moreover, we add a no-good-cut after three unsuccessful attempts to ensure that the same solution w.r.t. the simple states does not occur again and thereby avoid cycling between solutions.

Finally, we note that the solutions produced here are not necessarily optimal w.r.t. the model featuring the nonlinear equations (M). However, if no slack is needed and the objective function value is zero, then the corresponding solution is optimal.

3.9. Conversion of Natural Gas into Hydrogen Instances

In the context of repurposing existing natural gas transport infrastructure, one of our goals is to investigate whether such a network can be operated and how its control changes w.r.t. the transport of energy equivalent amounts of hydrogen. Therefore, and since all the data necessary to formulate our tri-level model for the natural gas transport network of OGE is available, we propose a method to convert it and derive meaningful hydrogen transport instances. The main issues concerning a transformation are the compression of hydrogen with turbo compressors and the necessary scaling of supplies and demands.

3.9.1. Hydrogen Compression with Turbo Compressor Units

First, we discuss whether and how the input parameters regarding an approximated compressor unit $m \in \mathcal{M}$ have to be changed when hydrogen instead of natural gas is transported. While the maximum available power $P_{m,t}$ remains the same, the maximum compression ratio $R_{m,t}$ and the maximum possible mass flow $Q_{m,t}$ must be adapted to reflect underlying physical and technical differences.

We start with a discussion on the maximum compression ratio. Recall that when gas is compressed by a turbo compressor machine, the connection between the pressure ratio, the amount of mass flow passing through, and the power necessary to realize it is described by the nonlinear power equation

$$\pi_{a,t} = \frac{q_{a,t}}{\eta_{\mathrm{ad}}} R_s T z_\ell \frac{\kappa}{\kappa - 1} \left[\left(\frac{p_{r,t}}{p_{\ell,t}} \right)^{\frac{\kappa-1}{\kappa}} - 1 \right], \tag{P}$$

where $\pi_{a,t}$ represents the power that is necessary to compress a mass flow of $q_{a,t}$ with pressure $p_{\ell,t}$ up to a pressure of $p_{r,t}$, see the discussion on the compressor arc model in Subsection 3.5.3 for more details.

Let us consider the equation above as a function of the compression ratio variable $R_{m,t} := p_{r,t}/p_{\ell,t}$ by fixing the mass flow value. For small values of $R_{m,t}$, in particular $R_{m,t} \in [1, 2.5]$, which is a typical range for the maximum compression ratio of a single turbo compressor regarding natural gas, we apply the following approximation

$$R_{m,t}^{\frac{\kappa-1}{\kappa}} = e^{\ln(R_{m,t}^{\frac{\kappa-1}{\kappa}})} = e^{\frac{\kappa-1}{\kappa}\ln(R_{m,t})} \approx e^{\frac{\kappa-1}{\kappa}(R_{m,t}-1)}.$$

From the first-order Taylor series at $R_{m,t} = 1$ of this expression, we then derive

$$\pi_{a,t} = \frac{q_{a,t}}{\eta_{\text{ad}}} R_s T z_\ell (R_{m,t} - 1).$$

Next, for fixed mass flow and power values, we compare the compression ratios of a turbo compressor when hydrogen and natural gas are compressed. Indicating the corresponding parameters for hydrogen by $\mathtt{H_2}$ and natural gas by $\mathtt{NG}$, we obtain

$$R_{m,t}^{\mathtt{H_2}} = 1 + \frac{R_s^{\mathtt{NG}} z_\ell^{\mathtt{NG}}}{R_s^{\mathtt{H_2}} z_\ell^{\mathtt{H_2}}} (R_{m,t}^{\mathtt{NG}} - 1).$$

The specific gas constants for hydrogen and natural gas are $R_s^{\mathtt{H_2}} = R/M^{\mathtt{H_2}} = R/2.0$ and $R_s^{\mathtt{NG}} = R/M^{\mathtt{NG}} = R/17.8$, respectively, where R is the universal gas constant and $M^{\mathtt{H_2}}$ and $M^{\mathtt{NG}}$ denote the corresponding molar mass values. Note that we use an average value for Norwegian H-Gas for the latter. The approximation formulas for the compressibility factors both depend on the gas pressure, see the paragraph on pipelines in Subsection 3.5.2 for details. Additionally, the approximation formula of Papay [126] for natural gas also relies on the temperature. Thus, to derive a linear expression, we assume an operating pressure of 50.0 bar and a temperature of $T = 25.0\,^\circ\text{C}$, which results in compressibility factors of $z_\ell^{\mathtt{H_2}} = 1.03$ and $z_\ell^{\mathtt{NG}} = 0.91$. Thus, for our conversion, we apply the following formula for the maximum compression ratio of a turbo compressor machine when hydrogen is transported, which is based on the corresponding maximum compression ratio for natural gas only

$$R_{m,t}^{\mathtt{H_2}} = 1 + \frac{R_{m,t}^{\mathtt{NG}} - 1}{10}. \tag{3.59}$$

Next, we discuss the changes that are necessary w.r.t. maximum possible mass flow. The feasible operating range of a turbo compressor unit is usually described by a characteristic diagram. An example of a turbo compressor and natural gas from the book of Koch et al. [96] is shown in Figure 3.6. For more details regarding this topic, we refer to [96] and the paper of Odom et al. [119]. We see that the maximum flow, in this picture the volumetric flow, for the transport of natural gas is bounded by the so-called chokeline on the right. Choking is the phenomenon when the mach number, i.e., the ratio between the velocity of the gas and the speed of sound, reaches 1 in some parts of the unit during compression. In that case, no further compression is possible, the flow is said to be choked or stonewalled, and

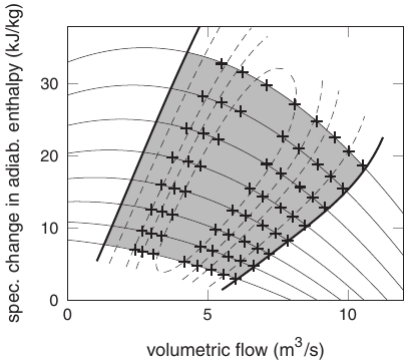


Figure 3.6.: Characteristic diagram of a turbo compressor for natural gas. Specific change in adiabatic enthalpy vs. volumetric flow rate: Dashed lines represent isolines for adiabatic efficiency, thin solid lines represent isolines for compressor speed. The left thick solid line represents the surgeline, the right thick solid line represents the chokeline. All curves are the result of least-squares fits with respect to measurements +.

prolonged operation of the compressor close to this limit can cause severe damage.

As the speed of sound in hydrogen $1,270\frac{\mathrm{m}}{\mathrm{s}}$ is much larger than the speed of sound in natural gas $446\frac{\mathrm{m}}{\mathrm{s}}$, this phenomenon does not limit the compression of hydrogen in practice. Considering the diagram in Figure 3.6, one can think of the chokeline being removed for hydrogen. However, here the maximum flow is limited by the maximum rotational speed. Experts at OGE estimate this value to be about 20% larger than the value imposed by the chokeline. Thus, we assume the relation

$$Q_{m,t}^{\mathtt{H_2}} = 1.2\, Q_{m,t}^{\mathtt{NG}}. \tag{3.60}$$

of the maximum mass flow values for natural gas and hydrogen in the following.

3.9.2. Energy Density

The second important aspect that we have to consider for our conversion is the energy density of hydrogen, which is much lower compared to natural gas. In particular, the amount of energy transported by natural gas is about 3.19 times bigger if we consider the same standard volumetric flows as supply and demands, since

$$3.19 \approx \frac{40.68\,\mathrm{MJ/m^3}}{12.75\,\mathrm{MJ/m^3}} = \frac{H_{s,n}^{\mathtt{NG}}}{H_{s,n}^{\mathtt{H_2}}}.$$

Here, $H_{s,n}^{\mathtt{NG}}$ and $H_{s,n}^{\mathtt{H_2}}$ denote an empirical calorific value for Norwegian H-Gas and hydrogen, see Cerbe [23], respectively. Thus, to evaluate the feasibility of a repurposed hydrogen transport network w.r.t. the energy it has to transport, we must adapt the boundary values accordingly.

Moreover, naturally, the original initial network state from natural gas for $t = 0$ does not account for the higher demands. Thus, in order to adapt and "recalibrate", e.g., w.r.t. the new compressibility factor, we do not fully scale up the supplies and demands starting at time step $t = 1$. In particular, we allow for some ramp-up time for the network here. Therefore, given the scaling factor of 3.19 and some $t_s \in \mathcal{T}$, we define

$$D^{\mathtt{H_2}}_{v,t} := D^{\mathtt{NG}}_{v,t} + \frac{\min\{t_s, t\}\,(3.19 - 1)}{t_s} D^{\mathtt{NG}}_{v,t} \quad \forall v \in \mathcal{V}^{\mathrm{b}},\, \forall t \in \mathcal{T}. \tag{3.61}$$

Thus, for the first t_s time steps, supplies and demands are linearly increased while they are fully scaled for the remaining time steps.

3.10. Computational Experiments

In this section, we present computational experiments that we conducted to test the suitability of our algorithmic approach for determining a stable transient control for gas transport networks. Therefore, we used 332 natural gas transport instances based on a major subnetwork of OGE's gas transport network and corresponding historically measured data as input. First, we checked the applicability of our model to natural gas transport. Afterward, by applying the methods proposed in the Section 3.9, we tested whether the converted network can transport energy equivalent amounts of hydrogen, investigated how the transport infrastructure needs to be changed to enable this, and compared how the network control differs.

3.10.1. Instances and Test Sets

We created six test sets for our experiments that we derived from 332 natural gas instances that OGE provided. The latter are based on one of their major subnetworks and corresponding historically measured pressure and flow data from two weeks in September. We considered 168 instances in 30-minute intervals starting at noon of virtual day one and ending at 11:30PM of virtual day four and 164 instances starting at midnight of virtual day six and ending at 09:30AM on a virtual day 9. The time difference of 30 minutes between two successive instances does not possess any meaning and is due to the data creation process.

The network, shown in Figure 3.7 and whose composition is summarized in the corresponding caption, is identical for all test sets. The properties of the seven main network stations, whose locations are indicated by blue circles, are stated in Table 3.1. In this context, Figure 3.5 in Section 3.4 shows network station E and the original intersection area located in the northwest. Finally, we note that five additional network stations represent single regulators from the original network.

All parameters and weights that were used in our models and algorithmic procedures are listed in Appendix A.1. In the following, we explicitly state the important ones w.r.t. the creation of our test sets and the analysis of the results.

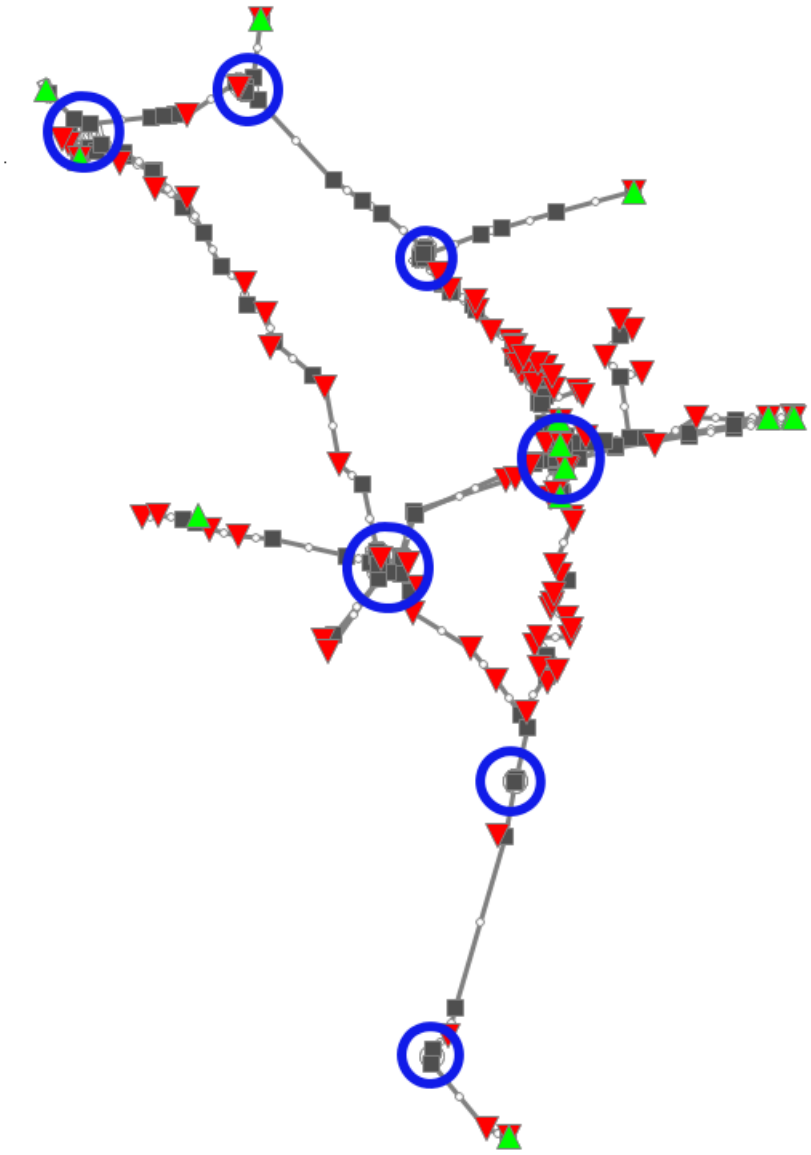


Figure 3.7.: The subnetwork of OGE's natural gas infrastructure used in our computational experiments. Inner nodes are shown as black squares, entries as green triangles ▲, and exits as red upside-down triangles ▼. All connections visible here represent pipelines. The locations of the seven main network stations A to G are highlighted using blue cycles. The network features $|\mathcal{V}| = 179$ vertices, $|\mathcal{V}^+| = 12$ entries, $|\mathcal{V}^-| = 89$ exits, $|\mathcal{A}^{\mathrm{pi}}| = 149$ pipes, and $|\mathcal{A}^{\mathrm{ar}}| = 72$ artificial arcs.

Name	$\lvert\mathcal{V}_i^{\mathrm{fn}}\rvert$	$\lvert\mathcal{V}_i^{\mathrm{ar}}\rvert$	$\lvert\mathcal{A}_i^{\mathrm{ar}}\rvert$	$\lvert\mathcal{A}_i^{\mathrm{co}}\rvert$	$\lvert\mathcal{F}_i\rvert$	$\lvert\mathcal{S}_i\rvert$
A	2	0	3	2	3	5
B	2	0	4	3	2	5
C	6	1	10	1	4	4
D	3	0	5	3	6	10
E	6	0	9	2	12	13
F	6	2	12	3	3	14
G	10	2	24	5	18	32

Table 3.1.: Overview of the properties of the seven main network stations A to G.

We used a temporal granularity of $2 \cdot 60$ minutes and $11 \cdot 120$ minutes for all instances, i.e., $k = 13$ time steps covering a time horizon of twenty-four hours. The six test sets differed in terms of the transported gas, the type of compressor units we considered, the supply and demand values and whether they have been balanced or not, and the inflow pressure bounds. For the sake of comprehensibility, the following descriptions of our six test sets are additionally summarized in Table 3.2.

The first test set consists of the original real-world instances provided by OGE. Natural gas is transported, and we used the given supplies and demands as boundary values for the entries and exits, the given inflow pressure bounds, and the turbo compressors with their associated parameters. We denote this test set by `NG-TC`.

Next, we created five test sets for hydrogen transport. Compared to natural gas, several parameters in our model need to be adapted, e.g., the compressibility factor or the isentropic exponent, see Section 3.5 on the tri-level MIP for more details.

For the first hydrogen test set, which we denote by `H2-TC`, we considered the turbo compressor units as provided in the original data. Hence, the maximum compression ratio and the maximum flow parameters regarding the machines were changed according to formulas (3.59) and (3.60), respectively. On the other hand, we did not apply the scaling procedure as suggested in Subsection 3.9.2. Hence, we used the same supplies and demands as in test set `NG-TC` in terms of volumetric flow, which means that only about one-third of the energy compared to natural gas is transported in the corresponding instances.

For the remaining four test sets, we replaced each turbo compressor unit with a special hydrogen compressor unit. We assume that each hypothetical hydrogen compressor units maintains the same maximum compression ratio, maximum power, and maximum flow as the corresponding original turbo compressor unit admits for natural gas. According to a whitepaper of several companies involved in gas transport [2], compressor units like this will be available when needed, i.e., when hydrogen transport networks are implemented.

Furthermore, for all four test sets, we applied the scaling procedure (3.61) as described in Subsection 3.9.2 to the boundary values of the underlying natural gas data to consider the same supplies and demands in terms of energy compared to `NG-TC`. In particular, since we use $t_s = 8$ as parameter in (3.61), after linearly

Test Set	Gas	Compressor Type	Scaling	Balancing	Relaxing
`NG-TC`	NG	TC			
`H2-TC`	H2	TC			
`H2-HC-EQ`	H2	HC	X		
`H2-HC-EQ-B`	H2	HC	X	X	
`H2-HC-EQ-P`	H2	HC	X		X
`H2-HC-EQ-B-P`	H2	HC	X	X	X

Table 3.2.: Summary of the properties of our six test sets. The first column denotes the names of the test sets, which do also encode the corresponding properties. The second column shows the type of gas that is transported, i.e., NG = natural gas or H2 = hydrogen. The third column denotes the type of compressor unit that we consider, i.e., TC = turbo compressor units or HC = hydrogen compressor units. Hence, for all hydrogen test sets, except `H2-TC`, we use the same machine parameters as the turbo compressor machines admit for natural gas in test set `TC-NG`. For `H2-TC`, the parameters are adapted according to (3.59) and (3.60). In the fourth column, an X shows whether the boundary values were scaled according to (3.61), which is encoded by `-EQ` in the test sets' names, too. Similarly, the last two columns indicate whether the balancing procedure (`-B`) for the boundary values and the relaxing of the inflow pressure bounds (`-P`) as explained in Subsection 3.10.1 are applied or not, respectively.

scaling up the boundary values for the first twelve hours. For the last twelve hours, the supplies and demands are equal in terms of energy.

The second hydrogen test set, `H2-HC-EQ`, features the original natural gas input using the hydrogen compressor units as well as the scaled boundary values.

We consider the same setup for the third hydrogen test set as for H2, but we additionally balanced supply and demand. To do this, we added up all the supply on the one hand and all the absolute demand on the other hand over the whole time horizon. Afterward, we evenly scaled up the boundary values of the entries or exits, depending on which one yielded the smaller sum, such that both sums coincided in the end. This test set we denote by `H2-HC-EQ-B` in the following.

Fourth, we again considered the same setup as for `H2-HC-EQ`, but we additionally relaxed the inflow pressure bounds by adding one bar to all upper and subtracting one bar from all lower bounds. Hence, for this test set, which we call `H2-HC-EQ-P`, we allow a broader range of future pressure values at the network's entries.

Finally, for the fifth hydrogen test set, we again took `H2-HC-EQ` as the basis, and applied both of the previously described modifications, i.e., we balanced supply and demand as in test set `H2-HC-EQ-B` and changed the inflow pressure bounds as in test set `H2-HC-EQ-P`. The resulting combination we denote as test set `H2-HC-EQ-B-P`.

3.10.2. Computational Setup

We performed our computations on a cluster of machines composed of two *Intel Xeon Gold 5122* running at 3.60 GHz, which provide in total eight cores and 96 GB of RAM. As solver for the underlying MIP problems, we used *Gurobi* version 9.1.2 [69], accessed via the C interface. Since the MIP and LP models turned out to be numerically challenging, we set the *NumericFocus* parameter to maximum value 3, the *IntegralityFocus* parameter to 1. Otherwise, we used standard settings. In this context, we also fixed the absolute velocities in the friction-based pressure difference term of the Momentum Equation (3.5), i.e., in the third summand, to the maximum of the absolute gas velocity at the corresponding node and $v^{\min} := 10^{-1}\frac{m}{s}$. The introduction of the threshold $v^{\min}$ is necessary to control the magnitudes of the constraint's coefficients and avoid numerical instabilities. This threshold is decreased to $10^{-3}\frac{m}{s}$ for the IVAP as the fixation of the binary variables allows for this more accurate modeling.

Next, we applied the rolling horizon heuristic with backtracking presented in Subsection 3.6.1 before each MIP solve to find an incumbent solution.

Finally, we set a cumulative time limit for each instance and all MIP solves, including the heuristic, of twenty-four hours. Additionally, we set individual time limits for the single-level MIPs of six hours, and for the MIP solves within the rolling horizon heuristic we imposed an additional time limit of 300 seconds. Further, besides the Δ-criterion, Algorithm 4 was terminated if the cumulative run time exceeded 24 hours, which was checked after each IVAP run. For the IVAP itself, no time limit was imposed.

3.10.3. Results

We split the presentation of the results of our computational experiments into three parts. First, we analyze the applicability of our algorithmic approach to natural gas transport. Thereby, we concentrate on the properties and the quality of the obtained solutions and on the run times, as these are essential factors concerning the underlying application, i.e., the usage within KOMPASS.

Afterward, we turn to the results for our hydrogen test sets. The main focus here is to check whether converting the natural gas infrastructure into a hydrogen transport network is feasible w.r.t. the network control. In particular, we are interested in whether there exist solutions without the usage of slack for the corresponding test instances or not. We also investigate what changes to the network and regulations are necessary to admit such a control in this context.

Finally, we compare the control of the network w.r.t. the two different gases. In particular, we analyze how much the necessary amounts of compression energy differ and how many technical measures were applied for both.

Results for Natural Gas

First, we focus on the performance of our algorithmic approach w.r.t. natural gas transport, i.e., we take a closer look at the results for test set `NG-TC`. The detailed results of these experiments can be found in the tables in Appendix A.2.

For all instances, a feasible tri-level MIP solution without slack was determined. Since we used the instances provided by OGE, which are based on historically measured data, we know that the initial state, future boundary values, and inflow pressure bounds correspond to network states that have occurred in practice. Furthermore, non-technical control measures, which the dispatchers may have applied within the considered time horizon, are implicitly included in the input data. Thus, if our model and the network stations capture reality well, we expect such a result.

Additionally, except for instance 6-2000, in which two simple state changes were performed, at most one was necessary for all the other instances. In particular, we observe that one technical measure was applied in the solutions for 40 instances, while in the remaining 291, none was necessary. These observations are in line with the expectations of the practitioners, who consider these results meaningful w.r.t. the test instances and the objective function of the third level in particular. Hence, as a first result, we deduce that our modeling approach is applicable here.

Next, Algorithm 4 terminated with zero optimality gap in its last tri-level MIP solve for all but nine instances, which also featured the longest run times. The latter highly depend on whether a simple state change was applied in the corresponding solution or not: Among the 311 instances, which had a total run time of less than one hour, are only 21 for which at least one simple state change was performed. On the other hand, no simple state change was conducted for only one of the 21 instances with longer run times. It is not surprising that those instances for which no technical measure was applied were often solved faster since they have zero as third-

level objective value, which is a trivial lower bound on the corresponding objective function. Additionally, a look into the logfiles of the MIP runs reveals that most of these solutions as well as most of the final (incumbent) solutions of the other instances were obtained early on in the solving process. For example, the preceding rolling horizon heuristic determined the final solutions for 295 instance. Hence, we can conclude that our algorithm can determine solutions of good quality in short amounts of time.

Finally, more than one iteration of Algorithm 4 was performed for fifteen instances, and two no-good cuts were added in total. The IVAP needed at most twelve iterations for each of its runs, i.e., it never hit the iteration limit Δ, and terminated successfully in all of its last runs except for instances 6-1200 and 6-1230, which both feature a high final MIP gap. Thus, the physical accuracy of the solutions could be improved for nearly all instances and in particular if the considered initial solutions were proven optimal for the corresponding tri-level MIP. Since this was often immediately possible, i.e., within the first iteration of Algorithm 4, the smoothed solutions of the tri-level MIPs, which feature a linear gas flow model, are a reasonable basis for this postprocessing routine. These results underline that the recommended measures determined with tri-level MIP can be considered a meaningful technical control.

Hydrogen Results

Next, we analyze the results for our hydrogen test sets. Note that in none of the solutions for the corresponding instances, inflow pressure slack was applied. Thus, if we say slack in the following, this always refers to boundary flow slack.

We start our discussion by comparing the results of test sets `H2-TC` and `H2-HC-EQ`, whose details can be found in the tables in Appendix A.3 and Appendix A.4, respectively. We observe that 281 instances in `H2-TC` needed boundary flow slack, while the same holds for only 106 instances in `H2-HC-EQ`. Moreover, the average amount of slack needed for the instances in `H2-HC-EQ` is only about 20% compared to `H2-TC`, although the supplies and demands were scaled up here. The comparatively vast amounts of slack in `H2-TC` are probably due to the reduced maximum compression ratios according to formula (3.59). This bottleneck does not affect the results of all other hydrogen test sets, as we considered special hydrogen compressor units here, i.e., the formulas proposed in Subsection 3.9.1 were not applied. Hence, the current turbo compressors units seem not suited for hydrogen transport, even if we consider the same supplies and demands in terms of volumetric flow as for natural gas in `NG-TC`.

Next, we turn to the results for test set `H2-HC-EQ`. To figure out what needs to be changed to enable a network control without slack, we analyzed the reasons for its usage. The first exciting relation we identified was the flow imbalance, which we define as the difference between an instance's total supply and its absolute demand

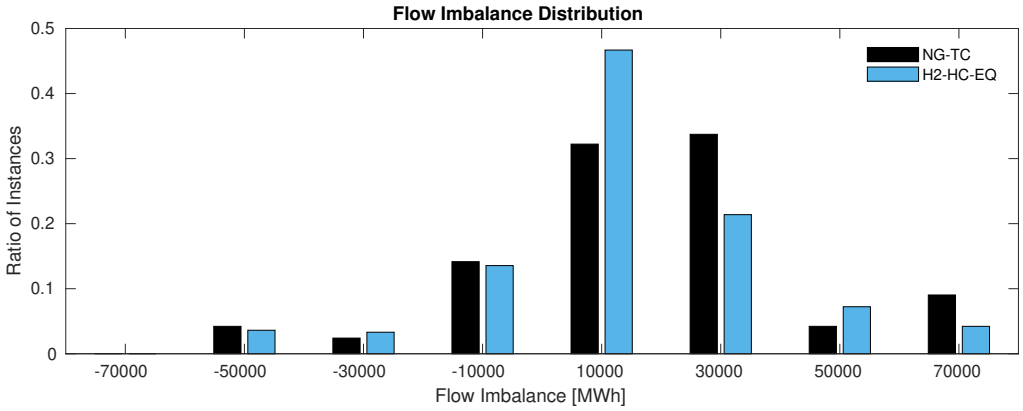


Figure 3.8.: Distribution of flow imbalances in MWh for test sets `NG-TC` and `H2-HC-EQ`. The instances are divided into corresponding intervals in this diagram, and the x-axis ticks depict their centers. On the other hand, the ratio of the contained instances is represented on the y-axis.

in terms of power over time (energy), i.e.,

$$\left(\sum_{t\in\mathcal{T}}(\tau(t)-\tau(t-1))\left(\sum_{v\in\mathcal{V}^+}D_{v,t}+\sum_{v\in\mathcal{V}^-}D_{v,t}\right)\right)H_{s,n}\,\rho^0 \tag{3.62}$$

where $H_{s,n}$ is the calorific value and ρ^0 the norm density of the considered gas. Recall that $\tau(k)$ is the length of the considered time horizon. Figure 3.8 shows a histogram of the flow imbalances in test sets `NG-TC` and `H2-HC-EQ`. The distributions here differ due to the applied ramp-up procedure (3.61). We observe that many instances feature a positive flow imbalance, which can be because of the following factors: First, the real-world natural gas data does not account for fuel gas used for compression. Second, regulation only demands that supply and demand must be balanced within the time window from 6AM to 6AM the other day. However, in practice, this rule is not strictly enforced by the TSOs for operational reasons. Finally, although we performed several consistency checks, we cannot guarantee perfect accuracy of the input data, as it originates from real-world measurements.

The relation between the flow imbalance of an instance in `H2-HC-EQ` and the total boundary flow slack used in the corresponding solution is shown in the scatter plot in Figure 3.9. We observe that the amount of slack is typically small and independent of the flow imbalance values in the range of -40,000 to 40,000 MWh. However, we also observe that the necessary slack scales linearly starting from an imbalance of about 40,000 MWh. Here, the inserted amounts of gas cannot be transported away from the entries. If no slack is applied, the pressure in the incident pipelines would increase and eventually violate the inflow pressure bounds.

To check whether the flow imbalance is the only reason for the usage of slack, we conducted the experiments with `H2-HC-EQ-B`. Recall that these instances are equiva-

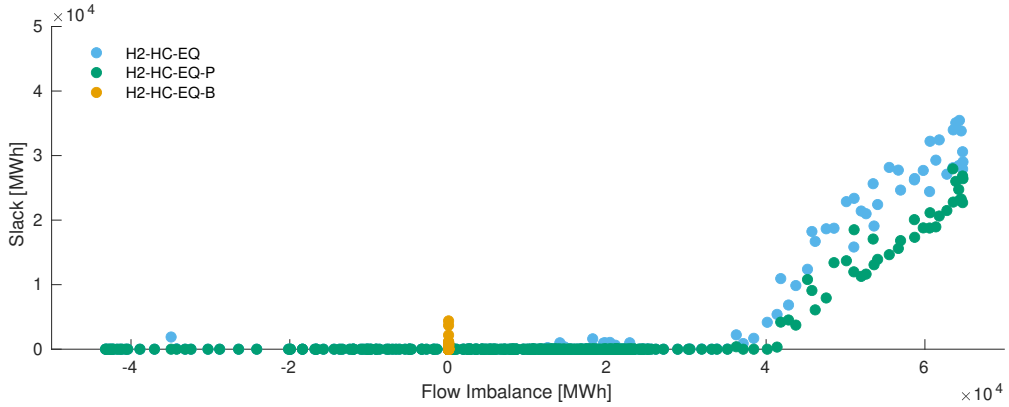


Figure 3.9.: Scatter plot displaying the boundary flow slack in MWh (y-axis) with respect to the flow imbalance in MWh (x-axis) for each instance of `H2-HC-EQ` (blue circles), `H2-HC-EQ-P` (green circles), and `H2-HC-EQ-B` (orange circles).

lent to `H2-HC-EQ`, but the boundary values were scaled up such that the total supply and absolute demand are balanced over time, i.e., there is no flow imbalance. Considering the results, which can be found in Appendix A.5, there are still 96 instances for which slack is needed. However, the corresponding amounts are relatively small compared to `H2-HC-EQ`, as we see in Figure 3.9. An analysis of the corresponding solutions revealed that most of this slack was still needed to decrease the supply at some major entries during the first time steps.

A plausible explanation for this behavior is that we start with an initial state originating from natural gas transport, and so do the considered future inflow pressure bounds. Therefore, when creating the instances, we implicitly assumed that hydrogen transport would also allow for or utilize such network states. Hence, we conjectured that this slack might be necessary for the network to adapt to the new medium and comply with the inflow pressure bounds.

Thus, to test our theory, we relaxed the inflow pressure bounds at the entries, which resulted in the creation of test set `H2-HC-EQ-B-P`. Detailed results of these experiments can be found in the tables in Appendix A.7. Here, all instances could be solved without any slack. To check whether the balancing is necessary and if the relaxation of the inflow pressure bounds alone suffices to achieve this goal, we created and ran test set `H2-HC-EQ-P`. The results, which can be found in Appendix A.6, show that this is not the case. There are still 38 instances for which significant amounts of boundary flow slack were needed. The corresponding relation with the flow imbalances is again shown in Figure 3.9. However, we observe that compared to test set `H2-HC-EQ`, the linear increase in the boundary flow slack is not as steep here.

Network Control – Natural Gas vs. Hydrogen

We base our analysis on comparing the network control w.r.t. the two different gases on the results of test sets `NG-TC` and `H2-HC-EQ`. Therefore, we considered all instances that finished with no slack and zero gap in the last MIP run for both test sets. In Figure 3.10, all 157 instances with this property are shown on the x-axis in chronological order. On the left y-axis, the amount of used compression energy is denoted. We calculated these values a posteriori for each instance using the correct nonlinear power equation for turbo compressor units (P) with the corresponding mass flow and pressure values. On the right y-axis, we denote the number of simple state changes that were conducted simultaneously with flow direction changes in the same network station. We use this measure for comparison instead of simple state changes only because the latter is often related to only switching on a compressor arc in a network station for `H2-HC-EQ`, which does not go along with a flow direction change. We do not consider those a genuine technical measure in our analysis as it does not generally change the overall network control.

First, with the increasing supplies and demands, more technical control measures become necessary for `H2-HC-EQ`. While for `NG-TC`, rarely one control measure needs to be applied, for `H2-HC-EQ`, we have one or two simultaneous simple state and flow direction changes for nearly all instances. An analysis of the solutions reveals that the increased need for technical control measures is due to the significantly reduced linepack compared to natural gas w.r.t. energy because of the lower energy density of hydrogen. Thus, less energy can be buffered and stored for future usage. In particular, while for instances with a positive flow imbalance, the pipelines in test set `NG-TC` can store the ingoing natural gas without the violation of any pressure bounds. This is an issue for the corresponding `H2-HC-EQ` instances, as discussed above. Here, the gas has to be distributed all over the network in order to store it. This explains the increased number of technical control measures for these instances, which transport the gas into network parts that were initially not supplied by the main entries. On the other hand, the reverse behavior can be observed in solutions of instances with a negative flow imbalance. Here the technical control measures are necessary to bring gas from initially separated network parts into circulation. Nevertheless, we note that the reduced linepack w.r.t. energy for hydrogen plays the decisive role in both cases.

Finally, a lot more compression energy, namely 440% more on average, is needed for the instances in `H2-HC-EQ` compared to `NG-TC`. On the one hand, this is due to the higher amounts of flow, which makes more compression necessary. On the other hand, as discussed above, the necessity to distribute the hydrogen in or collect it from different parts of the network additionally increases the need for compression.

3.11. Conclusion and Future Research

We split our conclusion into three parts. After discussing the applicability of our algorithmic approach to natural gas transport, we turn to the results regarding a

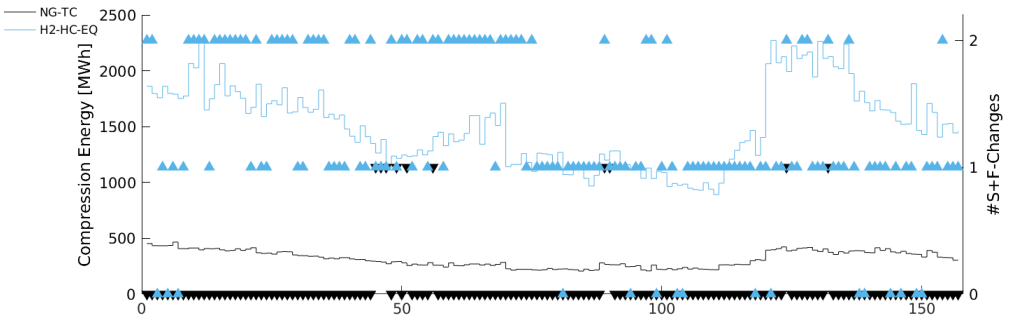


Figure 3.10.: Comparison of used compression energy and the simultaneous simple state and flow direction changes for all instances in `NG-TC` and `H2-HC-EQ` that finished with no slack and zero gap. On the x-axis, the instances are listed in chronological order. The colored curves together with the left-hand side y-axis show the amount of energy used for compression. The symbols scaled to the right-hand side y-axis show the number of simple state changes conducted simultaneously with flow direction changes.

potential conversion of the infrastructure into a hydrogen transport network. In both parts, we additionally discuss possible directions for future research. This chapter ends with some more general remarks regarding the experiments and comments on the idea of mixing gases with each other.

3.11.1. Natural Gas Transport

Our computational experiments, based on a major subnetwork of OGE's natural gas transport network together with historically measured flow and pressure data, demonstrate that our algorithmic approach represents a valuable basis for further development in KOMPASS. All test instances were solved without slack, and meaningful solutions were determined early in the solving process. For example, the preceding rolling horizon heuristic determined the final incumbent solutions for nearly 90% of the instances. Moreover, the number of recommended technical control measures is small, which is what practitioners expect considering the test set and the objective function of the third level of our tri-level MIP.

However, we also observe that there is much room for improvement, especially regarding the lower bound in the MIP solving process. The solver often spends much time on closing the gap for those instances where at least one simple state change is performed in the corresponding solution. To improve upon that, we are currently developing problem-specific inequalities, whose underlying idea is to prevent combinations of simple states and flow directions of "adjacent" network stations that are physically absurd or not meaningful in practice, e.g., two network stations compressing gas into opposite directions.

Additionally, as stated above, the considered test instances are based on historically measured supplies, demands, and pressure values. However, the model may need to apply non-technical measures, i.e., slack, when forecasted values are used. The latter can lead to a significant increase in run time, as the results for the hydrogen instances suggest.

From an application-oriented point of view, although the hand-tailored network stations that we used in our experiments work quite well, an important next step is creating an automatized process to derive these models based on the original topologies of the intersection areas and the corresponding operation modes. Such a routine would not only improve the accuracy of the overall model as it could be possible to show that the capabilities of each operation mode are included in the network station model, and we, therefore, consider a relaxation instead of an approximation here. It would also lead to a better maintainability of KOMPASS. Whenever the topology at a junction changes in the real world, the network station models currently have to be revised and adapted by hand, which is a time-consuming and resource-intensive task.

Furthermore, we aim at applying the model to larger parts of the network and thereby include more intersection areas. Of course, it is undoubtedly beneficial to extend our formulation, for example, by modeling parallel and sequential setting choices for compressor stations or to improve the solution quality by incorporating features like ramp-up times for compressor units, too. However, our current priority is to decrease the run time in order to be able to scale our algorithmic approach.

Finally, the whole procedure and the IVAP may benefit from another idea that we are currently pursuing: Exploiting the information from solutions of previously solved instances. In particular, consider some problem instance together with the chronologically preceding one. The former is usually based on the same network topology and features only slightly different input data compared to the latter. We are working on a heuristic that uses the solution values of the binary variables corresponding to the jointly covered time horizon. Moreover, we are also experimenting with using the absolute velocities from the solution directly in the Momentum Equations (3.5) of the initial tri-level MIP for the current instance. The rationale here is that these values may be a more realistic starting point than using the same absolute velocity for all time steps.

3.11.2. Hydrogen Transport

Our experiments on whether the underlying natural gas transport network can be converted into a pure hydrogen transport network yield promising results and give a positive answer provided that certain technical and regulatory conditions are met. However, before stating a conclusion here, we add a disclaimer.

To the best of our knowledge, there is no data regarding real-world, large-scale hydrogen transport networks since such an infrastructure does not exist yet. The instances that we used in our experiments are based on measured data from a real-world natural gas transport network and we converted them in a way that we con-

sider reasonable. However, as all the network states originate from natural gas transport, we implicitly impose that hydrogen transport allows for or even utilizes them in everyday operation, which is unlikely. Nevertheless, our study on whether the natural gas transport network can be repurposed reveals several exciting insights.

First of all, without a replacement of the currently installed turbo compressor units by special hydrogen compressor units, transport cannot be realized as the results for test set `H2-TC` demonstrate. This is due to the maximum compression ratio of the turbo compressor units for hydrogen, which is drastically reduced compared to natural gas, see equation (3.59) in Subsection 3.9.1.

Additionally, our analysis regarding the remaining test sets shows that transport is possible in case that supply and demand are balanced over the considered time horizon. The main reason for this being that, due to the lower energy density of hydrogen, the linepack w.r.t. energy is significantly reduced compared to natural gas. Hence, less energy can be stored or retrieved for future usage. Therefore, it seems necessary to enforce stricter regulatory rules in a future hydrogen transport market, e.g., to require the traders to balance supplies and demands within twelve or fewer hours compared to the twenty-four currently established in natural gas transport.

For the same reason, more technical control measures become necessary. For `H2-HC-EQ`, at least one simultaneous simple state and flow direction change is performed for nearly all instances, while for `NG-TC` none is conducted in a majority of the scenarios. The necessity to distribute the hydrogen in or collect it from more remote parts of the network, depending on whether we have a surplus in supply or demand, makes this more dynamic control inevitable.

Furthermore, besides the fact that more than three times the amount of volumetric flow is transported through the network in `H2-HC-EQ`, the changes in the control add to the increased need for compression energy. In particular, we find that 440% more compression energy is needed on average compared to natural gas test set `NG-TC`.

Summing up, repurposing the network seems possible if the turbo compressor machines are replaced with special hydrogen compressor units and stricter rules regarding the balancing of supply and demand are enforced. However, we expect that the amount of compression energy would drastically increase and that the network control becomes more dynamic.

Regarding the algorithmic approach itself, we are currently planning to use it to develop a decision support system for hydrogen transport, similar to KOMPASS. As the control of hydrogen networks is likely to be more dynamic, such a tool may be even more valuable to prospective dispatchers. However, we need to conduct a more intensive analysis of the balancing requirements and develop suitable pressure profiles that are feasible for hydrogen transport. The latter is of particular importance, as the practitioners at OGE assume that it is important to keep the pressure differences in such a network small over time. A control that does not respect this requirement may boost pipeline embrittlement. Thus, we are currently discussing corresponding changes in the objective functions of the tri-level MIP model for hydrogen transport.

3.11.3. General Remarks

As the Energiewende in Germany and worldwide progresses and decisions on implementing power-to-gas options are discussed intensively, there is a great need for future research regarding the control of prospective transport networks. Regarding the experiments presented here, we are working on extending them and want to use a wider variety of test instances in order to verify the obtained results. For example, as the data that we used stems from two weeks in September, which usually does not feature the highest annual supplies and demands, additional parallel hydrogen compressors may actually become necessary during wintertime.

Another critical topic is the mixing of gases. The amounts of produced chemical energy carriers such as hydrogen will initially be relatively small during an energy transition. Thus, no complete conversion of the gas networks is going to be necessary in the near future. Nevertheless, one could insert surplus hydrogen into the network and blend it with natural gas in an intermediate step. This can be done up to a ratio of 10% without having any significant impact on compression [2]. However, due to regulatory constraints, for example industries relying on a certain quality of natural gas, a practicable ratio will be smaller. A more detailed discussion on this topic as well as a first case study regarding the hydrogen capacity of the current natural gas network has been conducted by Pedersen et al. [127]. From a mathematical point of view, mixing gases adds another source of complexity to the problem of determining a transient network control. In particular, it becomes necessary to trace the gas composition, which is typically done using nonlinear pooling constraints.

4. Identifying Severe Transport Scenarios

In this chapter, we discuss the problem of identifying severe transport scenarios in natural gas transport. To understand how transport scenarios arise in the first place, we start with a description of the current natural gas market model, its regulatory rules, and the corresponding challenges that the transport system operators (TSOs) face in Section 4.1. Furthermore, we present several methods to determine and evaluate the technical capacity of a gas network, i.e., the maximum amounts of gas that can be injected into or withdrawn from the network while its technical control can still be guaranteed. In this context, we consider approaches from the literature as well as methods that are applied in practice. The main idea that the latter have in common is to decide on the feasibility of technical capacities by evaluating a finite set of severe transport scenarios using highly detailed mathematical models for the physics of gas transport. These test sets are generated and designed with the goal in mind to cover and represent all possible difficult flow situations that can arise. If transport can be realized for all or at least a great majority of these scenarios, the technical capacity is considered feasible.

However, even for practitioners, e.g., the dispatchers who control the gas networks and know them very well, it is hard to specify what "severe" means w.r.t. gas transport scenarios. We review existing severity measures and look at the approaches for generating related test sets. Afterward, we propose two new measures, namely the minimum transport moment and the potential transport moment, together with methods to identify corresponding worst-case scenarios. To this end, we introduce two new network flow problems: The Maximum Transportation Problem in Section 4.2, a generalization of the well-known Transportation Problem, and the Maximum Potential Transport Moment Problem in Section 4.3. Besides an analysis of their computational complexity, we propose heuristic and exact solution approaches for both. A case study based on the gaslib-582 instance from the GasLib benchmark library [147] is conducted in Section 4.4. The results confirm that our measures should be considered in the future since the transport scenarios determined with MaxTP and MaxPTM exceed the corresponding maximum severity values from a provided test set by 23% and 30%, respectively.

Note that this chapter focuses on natural gas as its transport in pipeline networks was established decades ago and the regulatory rules are clearly stated. However, since a similar market model will probably be implemented in a future hydrogen economy, the obtained results could be relevant in the context of its transport, too.

Finally, we believe that MaxTP and its generalization, the Maximum Minimum Cost Flow Problem (MaxMCF), which is discussed in Appendix B.1, constitute challenging new network flow problems for the scientific community on their own.

4.1. The Entry-Exit Gas Market Model

To determine severe transport scenarios for natural gas networks, we need to understand how these arise based on the current market model and the corresponding regulatory rules. Hence, this section discusses the essentials of natural gas trading in Germany and the challenges that the transmission system operators (TSOs) face. The following explanations are based on the book chapter of Rövekamp [140].

Traditionally, the same company provided natural gas trading and transport, which enabled long-term planning based on reliable forecasts of supply and demand. All necessary information for an efficient network control was available early on, and critical network situations could be avoided or resolved through internal communication. On the other hand, due to this organization, the gas customers, e.g., power plants, industrial sites, or utility networks, had no choice but to buy their gas from a company to whose transport network they were physically connected. Thus, the European Union decided to decouple trading and transport to implement a more liberal market structure according to a directive [42] and subsequent regulation [43]. However, this effort led to the establishment of more complex market models.

In Germany, the so-called entry-exit model was introduced in 2005. Here, for each market area, which represents a region or a part of the gas grid for which this model is implemented, a virtual trading point (VTP) is installed. This VTP incorporates all entry and exit points of the corresponding area. To inject gas into or withdraw it from the network, the transport customers, which we synonymously call shippers in the following, book capacity contracts for the desired entries or exits in an auctioning process. Although there exist capacity contracts that are closed the day before (day-ahead capacities) or even on the same day (intraday capacities), most of the bookings, which we focus on in this chapter, are realized several weeks or years before transport. Within their contractual rights, the shippers then nominate the actual amounts of gas that they want to transport the day before. Modifications of these nominations, i.e., renominations, are possible up to two hours before realization according to specific rules limiting the extent of deviations.

However, when making their nominations, the shippers all together have to ensure that supply and demand are balanced, i.e., that the amount of gas injected into the network equals the amount of gas withdrawn from it. This is necessary to allow for a stable network operation and is typically realized for each gas day, i.e., within the 24 hours between 6AM and 6AM the next day. To achieve this, the shippers trade with other transport customers having matching rights by buying or selling gas, which is virtually exchanged at the VTP. For the sake of understandability, we assume that all nominations correspond to freely allocable capacities here, i.e., there is no restriction on the entries or exits that the shippers can trade with. For more information on the topic and examples of other available, more restrictive capacity products, we refer to the thesis of Hayn [74]. The concepts of bookings and (balanced) nominations are illustrated in Figure 4.1.

Additionally, we distinguish two main types of capacity that are auctioned: Firm and interruptible capacities. In the case of firm capacities, the TSOs guarantee that

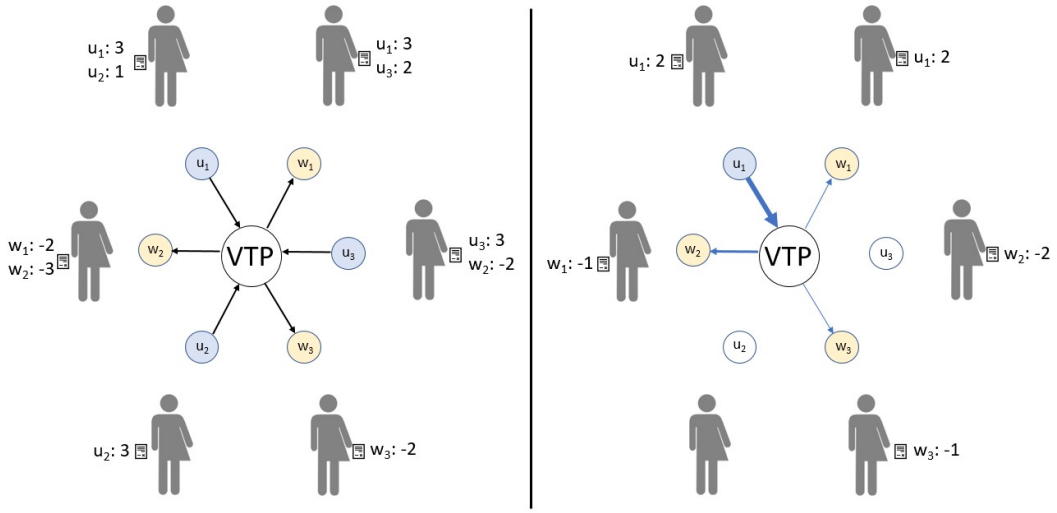


Figure 4.1.: On the left, a VTP with its entries u_1, u_2, and u_3 as well as its exits w_1, w_2, and w_3, together with six shippers is shown. Their individual bookings are stated directly next to them. On the right, we see a set of possible nominations. Note that supply and demand are balanced.

the corresponding nominated amounts of gas are transported except for situations considered as force majeure, e.g., severe technical network failures beyond the control of the TSOs. On the other hand, the TSOs can partially or entirely decline nominations corresponding to interruptible capacities if this is necessary to ensure a feasible network control. However, best endeavours have to be taken in order to realize them. Note that such interruptions are also mentioned and discussed in the context of non-technical measures for gas transport, see Section 3.3. Nevertheless, in the remainder of this chapter, we refer to the sum of all booked firm capacity contracts simply as a booking, and the sum of all nominations w.r.t. firm capacities is called a transport scenario or a supply and demand vector.

The TSOs are required by law to regularly determine and offer maximum amounts of firm capacity to the shippers, i.e., they need to determine the technical capacity of their network. In contrast to short-term agreements, this is more difficult when considering long-term contracts as forecasting future network states and nominations becomes harder, see the work of Petkovic et al. [130] as a reference for this claim.

Further, according to the Gas Network Access Ordinance (Gasnetzzugangsverordnung 2010, §9 (2)) [56], the corresponding evaluations have to be conducted using state-of-the-art techniques, including flow simulations, forecasting, and statistical methods. Thus, even the legislative authority explicitly acknowledges the need for mathematical methods to ensure the efficiency of the implemented entry-exit market model. The relevance of mathematical methods is further emphasized since the

interaction between the TSOs and the shippers, for example, can be described by a hierarchical four-level model, see Grimm et al. [64]. Based on this model, an analysis of the cost of the presented regulatory approach, i.e., the welfare losses due to unused network capacity, has been conducted by Böttger et al. [16].

Finally, three major mathematical problems of interest to us arise because of the entry-exit market model. They are, for example, stated and discussed in the paper of Fügenschuh et al. [49] together with their relations and interdependencies. While checking the feasibility of a transport scenario w.r.t. the (transient) control of the network has already been treated in Chapter 3, we review optimization approaches and complexity results regarding the other two in the remainder of this section. In particular, after discussing the problem of determining the technical capacity of a gas transport network, we review methods and results on deciding whether a given booking is feasible or not, i.e., if transport can be realized for all possible supply and demand vectors. Note that in contrast to Chapter 3, we focus on stationary networks in the following, since all previous work and our contributions consider no temporal resolution.

4.1.1. Technical Capacities

A first mathematical model for determining the technical capacity of natural gas networks is presented in the PhD thesis of Willert [161], which can be interpreted as an adjustable robust optimization problem. Furthermore, several properties of the problem are investigated, e.g., upper bounds on the maximum firm capacities of individual nodes are described. Moreover, the geometry of the set of feasible transport scenarios w.r.t. a nonlinear transport model is analyzed. In particular, the author shows that it can be nonconvex and may even contain holes using small example networks. Additionally, the important fact that the individual capacities of entry and exit nodes are in general not independent is discussed. For example, increasing the firm capacity of one entry may make it necessary to decrease it at another entry to avoid congestion in an adversarial transport scenario. Furthermore, it is emphasized that the overall problem itself allows for multiple interpretations as there is no clear definition of what maximum w.r.t. technical capacity means. For example, from the least restrictive point of view, the problem admits a set of Pareto-optimal solutions.

Another option to define the maximum is w.r.t. some weight vector on the entry and exit nodes, which is, for example, described in the technical report of Fügenschuh et al. [52]. For this variant of the problem, a solution approach consisting of two stages is introduced in the PhD thesis of Hayn [74]. In a first step, the set of bookings is successively split into boxes, and it is checked whether they contain any feasible supply and demand vector or not. These boxes are then marked as feasible and infeasible, respectively. Based on this information, maximum technical capacities are determined in a subsequent step. However, due to the complex geometry of the set of feasible transport scenarios mentioned above, feasibility can only be guaranteed up to a certain tolerance.

The maximization of a weighted sum of the firm capacities is also considered Schewe et al. [146]. In their article, the authors introduce three variants of the Computing Technical Capacities Problem (CTC) for passive networks, i.e., no active elements like compressors or valves are considered here. While the first variant features a capacitated linear flow formulation, the other two incorporate a linear and a nonlinear potential-based flow model, respectively. In particular, the nonlinear model features the well-known Weymouth pressure drop equation. The authors show that CTC is NP-complete on trees for all three variants using a reduction from SubsetSum, implying the NP-hardness for general networks.

4.1.2. Deciding the Feasibility of a Booking

After reviewing several approaches for determining the technical capacity of a gas transport network, i.e., to maximize the firm capacities at its entries and exits, we consider the decision variant of the problem in this subsection: Given a booking, is it feasible, i.e., does there exist a feasible technical control for all possible transport scenarios? We partition the corresponding solution approaches into two categories. First, we review what we call the complete methods, and as the name suggests, all possible nominations are checked for feasibility here. On the other hand, if the validation is based on evaluating a finite set of severe transport scenarios to obtain a representative and meaningful result, we call the approach sample-based.

Complete Approaches

A bilevel programming approach including active elements has been proposed by Plein et al. [133]. Here, the upper level chooses a feasible transport scenario, and the follower has to subsequently determine a feasible network control. Thereby, the lower-level gas transport model incorporates the Weymouth pressure drop equation and linearly modeled compressors and control valves. Additionally, to ensure the existence of a feasible solution, two slack variables on the pressure bounds of the network nodes are introduced. While it is the goal of the upper level to choose a scenario such that the sum of the slack variables is maximized, the lower level aims at minimizing it. If a solution's objective value is positive, there is no feasible network control without slack for the corresponding transport scenario. Further, suppose the active elements are not located on a cycle. In that case, the bilevel model can be reformulated as a single level problem using the classical KKT approach or one of three optimal value function reformulations. Computational experiments based on GasLib data [147] are presented, showing that the considered problem instances are numerically unstable. However, when replacing the Weymouth equations with a linear potential-based model, all of them are solved reliably and quickly.

Nevertheless, to the best of our knowledge, the approach described above is the only one considering active elements. For passive networks and potential-based flow models, conditions based on supply and demand vectors that induce maximum potential differences between the network nodes can be used to decide the feasibility

of the corresponding booking, see Labbé et al. [101]. Due to this result, it can be shown that the feasibility of a booking can be checked in polynomial time for the linear potential-based flow model and that the same holds true for nonlinear potential-based flows on tree networks. Additionally, using techniques from real algebraic geometry, Labbé et al. [102] show the same result for single cycle networks.

Moreover, a summary of several complexity results for passive networks is given in the paper of Labbé et al. [101], too. Here, the problems are classified w.r.t. the network structure, i.e., tree-shaped and cyclic networks, and the underlying gas flow model, i.e., capacitated and uncapacitated linear, linear potential-based, and nonlinear potential-based formulations. While most problems are in P, deciding the feasibility of a booking for cyclic networks and the capacitated linear flow model is, for example, coNP-complete, which is proven in the thesis of Hayn [74] using a reduction from GapExpansion.

Sampling-Based Methods

The approaches described above consider all booking-compliant supply and demand vectors and implicitly check them for feasibility. However, when applying more detailed models for the physics of gas transport, e.g., formulations including more accurate and typically nonlinear descriptions of gas compression or valves, which usually lead to the introduction of binary variables, corresponding formulations would become computationally intractable. In this case, the commonly used approach is to base the validation of a booking on the evaluation of a finite set of transport scenarios. In this context, it is crucial to determine difficult and severe transport scenarios to obtain a representative and meaningful result.

One approach, which has been developed within the research cooperation "ForNe" between the German TSO Open Grid Europe GmbH and several research institutions, is documented in the book of Koch et al. [96]. Thereby, historical data for gas demand at the exits are used to estimate distributions, which are subsequently used to sample realistic scenarios. Each scenario is completed by supply values that are chosen to be extreme, meaning that the scenario is balanced and maximal w.r.t. a random preference order on the entries. Afterward, the numerous resulting transport scenarios are individually checked for feasibility in an automated fashion using different optimization methods, including detailed MINLP and NLP models.

On the other hand, the TSO Gasunie from The Netherlands applies a slightly different approach, described in the paper of Steringa et al. [153]. First, a set of transport scenarios, which they call "stress tests", is identified. Afterward, these stress tests are validated using simulation tools, in particular the Multi Case Approach (MCA) presented in the PhD thesis of van der Hoeven [81]. If all of them admit a feasible network control, the corresponding booking is considered feasible.

The underlying idea of the approach is to determine stress tests that are maximally severe w.r.t. the topology of the network. Thereby, the severeness is quantified using the *transport moment*, a measure of flow transported over long distances. Additionally, all relevant directions of flow shall be covered. The generation procedure

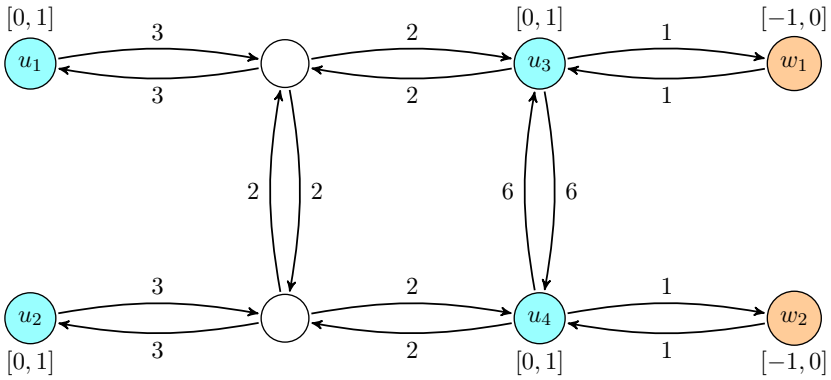


Figure 4.2.: Example network with cyan entries $\mathcal{V}^+ = \{u_1, u_2, u_3, u_4\}$ and orange exits $\mathcal{V}^- = \{w_1, w_2\}$. The arc lengths and the feasible supply and demand intervals are located next to the corresponding entities.

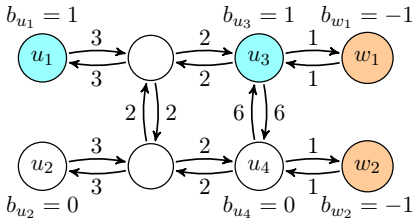
for the stress tests is based on distances relative to a reference point and is explained in more detail in the upcoming paragraph. Finally, the size of the resulting test set is reduced using a similarity measure, which is described in the Master thesis of Lindenberg [106].

The Reference Point Method

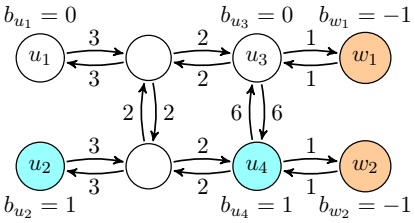
The Reference Point Method (RPM) presented by Steringa et al. [153] is an algorithm that attempts to find supply and demand vectors that maximize the transport moment $T = QD$, which is defined as the product of the system throughput Q, i.e., the total flow from the entries towards the exits, and the so-called mean transportation distance D. To determine it, a reference point in the network is chosen, and the distances to all entry and exit nodes are determined. Afterward, D is defined as the difference of the centers of gravity of the entry and exit nodes, which are the supply- and demand-weighted average distances towards the reference point.

In the following, we discuss the version of the RPM in more detail, which, according to [106] and [153], is used in practice to determine the set of stress tests, i.e., transport scenarios which are severe w.r.t. the transport moment. Therefore, we consider a connected flow network $G = (\mathcal{V}, \mathcal{A})$ with arc lengths $\ell_a \in \mathbb{R}_{\geq 0}$. Further, for each arc $(u, v) \in \mathcal{A}$, there exists an antiparallel arc $(v, u) \in \mathcal{A}$ having the same length, i.e., $\ell_{uv} = \ell_{vu}$. This is because passive networks are considered here where the flow in the pipelines can go in both directions.

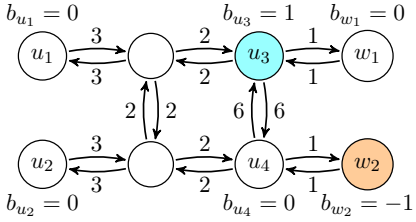
Next, a boundary node $v \in \mathcal{V}^+ \cup \mathcal{V}^-$ is chosen as reference point, and the lengths of shortest paths between it and all entries and exits are determined, i.e., d_{uv} for all $u \in \mathcal{V}^+$ and d_{vw} for all $w \in \mathcal{V}^-$, using, for example, Dijkstra's algorithm [34]. Additionally, for each entry $u \in \mathcal{V}^+$, we are given its maximum possible supply $\bar{b}_u \in \mathbb{R}_{\geq 0}$, and the maximum demand of each exit $w \in \mathcal{V}^+$ is given as $\underline{b}_w \in \mathbb{R}_{\leq 0}$. An example network illustrating this setup is shown in Figure 4.2.



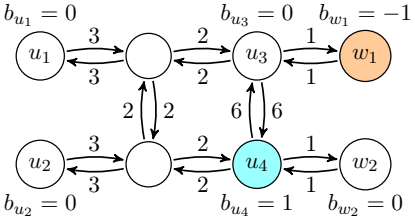
(a) Using u_1 as reference point.



(b) Using u_2 as reference point.



(c) Using u_3 or w_1 as reference point.



(d) Using u_4 or w_2 as reference point.

Figure 4.3.: The set of global stress tests for the example network in Figure 4.2. Entries and exits with non-zero supply and demand are highlighted.

RPM starts with the zero flow transport scenario, i.e., all supply and demand values are equal to zero. Next, the supply at an entry, which is not at its maximum and closest to v, and the demand at an exit, which is not at its maximum and farthest from v, are increased until one of them hits its bound. The procedure is repeated until an increase of the flow leads to a decrease in T. This happens when the centers of gravity shift, that is, the average distances between entries and exits are reduced. At this point, a stress test is found, and the algorithm terminates returning the current supply and demand vector.

In other words, the transport moment can be described as a linear function in supply and demand variables $b_v \in [0, \bar{b}_u] \subseteq \mathbb{R}_{\geq 0}$ and $b_w \in [\underline{b}_w, 0] \subseteq \mathbb{R}_{\leq 0}$ for all $u \in \mathcal{V}^+$ and $w \in \mathcal{V}^-$, respectively, with the condition that the resulting transport scenario it balanced. Hence, the algorithm solves the following linear program

$$
\begin{aligned}
\max_{b} \quad & -\sum_{u \in \mathcal{V}^+} d_{uv} b_u - \sum_{w \in \mathcal{V}^-} d_{vw} b_w \\
\text{s.t.} \quad & \sum_{u \in \mathcal{V}^+} b_u + \sum_{w \in \mathcal{V}^-} b_w = 0 \\
& b_u \in [0, \bar{b}_u] \quad \forall u \in \mathcal{V}^+ \\
& b_w \in [\underline{b}_w, 0] \quad \forall w \in \mathcal{V}^-.
\end{aligned}
$$

The transport scenarios generated by running the algorithm for each $v \in \mathcal{V}^+ \cup \mathcal{V}^-$ are called *global stress tests*. The six resulting global stress tests for the example

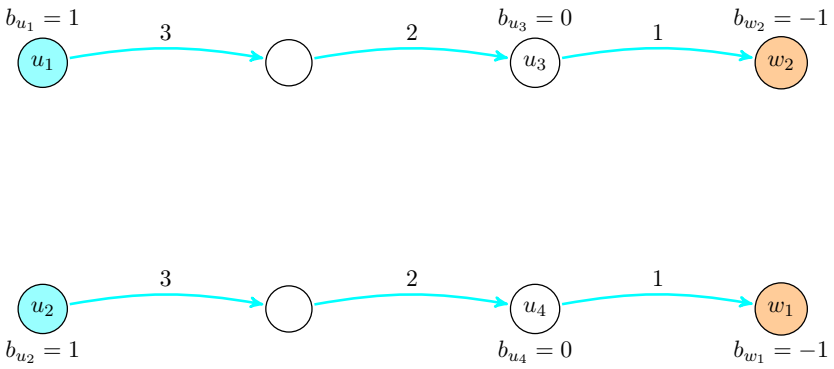


Figure 4.4.: This picture shows an example transport scenario that RPM does not find. Additionally, a minimum cost flow for the considered supply and demand vector is shown. While all cyan arcs carry one unit of flow, the invisible arcs carry no flow.

network in Figure 4.2 are shown in Figure 4.3.

However, as in the example, especially if multiple entries and exits are located close to each other, it can happen that some of the global stress tests, which we derive when using them as reference points, are identical. A set of reference points leading to the same global stress test is called a cluster. There are two clusters in our example: The sets $\{u_3, w_1\}$ and $\{u_4, w_2\}$. By maximizing the transport moment within the single clusters, additional so-called *local stress tests* are derived. For more details, we refer to the paper of Steringa et al. [153].

Nevertheless, due to the fixation of the distances towards the reference point and the presence of a cycle in our example network, the transport scenario shown in Figure 4.4 is not found by the algorithm. Although Steringa et al. [153] mention a procedure based on the clusters to handle such issues, it is unclear how this particular result can be derived.

On the other hand, consider the optimal objective values of the Transportation Problem instances induced by the supply and demand vectors of the six stress tests and the example in Figure 4.4. While the former yield nine or seven, the latter maximizes this value among all considered transport scenarios with a cost of twelve. This demonstrates that the transport scenario from Figure 4.4 should be considered severe, too, which also becomes evident from a visual inspection.

This observation motivates the definition of a new severity measure for gas transport scenarios: The *minimum transport moment*, i.e., the objective value of the induced Transportation Problem instance. In this context, we introduce the *Maximum Transportation Problem* (MaxTP) in the upcoming Section 4.2, which is a new generalization of the Transportation Problem. Its goal is to identify a supply and demand vector having a maximum minimum transport moment.

4.2. The Maximum Transportation Problem (MaxTP)

This section considers a new severity measure for transport scenarios, namely the minimum transport moment, i.e., the optimal objective value of the induced Transportation Problem (TP), which has been motivated and derived at the end of the previous section. In particular, we are interested in finding the most severe supply and demand vectors w.r.t. this measure. Thus, in the following, we introduce the Maximum Transportation Problem (MaxTP).

For MaxTP, we mainly consider the same setup as for general TP, see Section 2.3.1. We are given a flow network $G = (\mathcal{V}, \mathcal{A})$ with entries $\mathcal{V}^+ \subseteq \mathcal{V}$ and exits $\mathcal{V}^- \subseteq \mathcal{V}$, and we assume w.l.o.g. that $\mathcal{V}^+ \cap \mathcal{V}^- = \emptyset$. Furthermore, the remaining nodes $\mathcal{V}^0 := \mathcal{V} \setminus (\mathcal{V}^+ \cup \mathcal{V}^-)$ are the inner nodes. Additionally, for each arc $a \in \mathcal{A}$ we are given a nonnegative length value $\ell_a \in \mathbb{R}_{\geq 0}$. However, in contrast to TP, for MaxTP we are given feasible intervals for the supply and demand values. In particular, for each entry $u \in \mathcal{V}^+$, we have a nonempty supply interval $[\underline{b}_u, \bar{b}_u] \subseteq \mathbb{R}_{\geq 0}$, where $\underline{b}_u \in \mathbb{R}_{\geq 0}$ is a lower and $\bar{b}_u \in \mathbb{R}_{\geq 0}$ is an upper bound on its supply. Analogously, for each exit $w \in \mathcal{V}^-$, we have a nonempty demand interval $[\underline{b}_w, \bar{b}_w] \subseteq \mathbb{R}_{\leq 0}$ with bounds $\underline{b}_w, \bar{b}_w \in \mathbb{R}_{\leq 0}$. We denote an instance of MaxTP as quintuple $\mathcal{I} = (\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$.

Next, $b \in \mathbb{R}^{|\mathcal{V}^+ \cup \mathcal{V}^-|}$ is called a supply and demand vector or transport scenario. Furthermore, it is called admissible if $b_v \in [\underline{b}_v, \bar{b}_v]$ for all $v \in \mathcal{V}^+ \cup \mathcal{V}^-$ and $\sum_{v \in \mathcal{V}^+ \cup \mathcal{V}^-} b_v = 0$, i.e, if supply and demand respect the interval bounds and are balanced.

A solution for a MaxTP instance $\mathcal{I}$ is a tuple (b, f), where b is an admissible supply and demand vector and f is a feasible solution for the TP instance induced by b. The cost of a solution $c(b, f)$ is equal to the cost of the flow, i.e., $c(b, f) := \sum_{a \in \mathcal{A}} \ell_a f_a$.

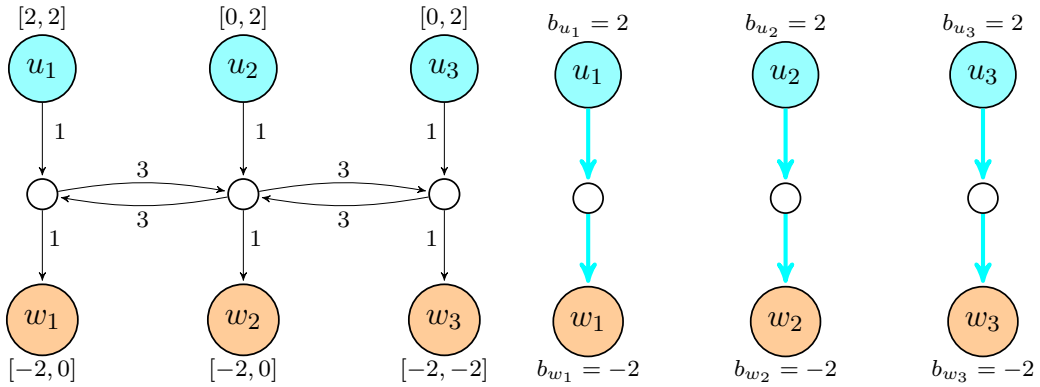
Moreover, a solution (b, f) is called feasible if f is optimal for the TP instance induced by b, and we call the corresponding objective value $c(b, f)$ the minimum transport moment of the transport scenario b.

The goal of MaxTP is to find an admissible supply and demand vector such that the minimum transport moment is maximized. Thus, a solution (b, f) is optimal for MaxTP if it is feasible and if $c(b, f) \geq c(\tilde{b}, \tilde{f})$ holds for all feasible solutions $(\tilde{b}, \tilde{f})$. An example instance with three feasible solutions is shown in Figure 4.5.

Similar to Ahuja et al. [4], we assume w.l.o.g. that there exists a directed path from each entry towards each exit in the network to ensure the existence of a feasible solution for each TP instance induced by an admissible supply and demand vector. We impose this connectedness-condition, if necessary, by adding direct arcs with big length. No such arc appears in a feasible solution for MaxTP unless there exists an admissible supply and demand vector inducing an originally infeasible TP instance.

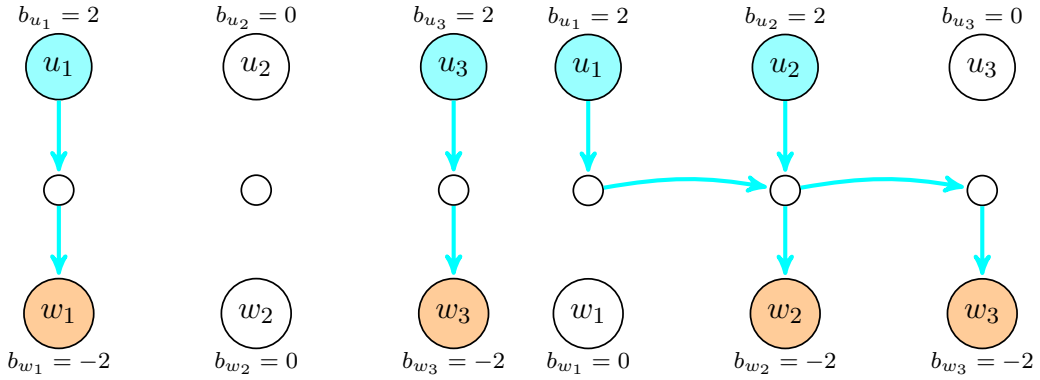
4.2.1. Optimal Solution Structure

Next, we prove an interesting result regarding the structure of an optimal solution for MaxTP. Therefore, we introduce the notion of bound-closeness for admissible supply and demand vectors.



(a) Flow network with $\mathcal{V}^+ = \{u_1, u_2, u_3\}$ and $\mathcal{V}^- = \{w_1, w_2, w_3\}$. The supply and demand intervals are shown above and below the nodes, and the length values located next to the corresponding arcs.

(b) A feasible solution featuring the maximum supply of six units. The optimal solution of the induced TP instance shown here yields the corresponding minimum transport moment of $c(b, f) = 12$.



(c) A feasible solution featuring four supply units with a minimum transport moment of $c(b, f) = 8$.

(d) An optimal solution for the MaxTP instance with four supply units and a minimum transport moment of $c(b, f) = 20$.

Figure 4.5.: While Figure 4.5a shows an example MaxTP instance, Figures 4.5b, 4.5c, and 4.5d depict feasible solutions for it. The supply and demand values chosen in the latter are stated below or above the corresponding nodes. Furthermore, the blue arcs have a flow value of $f_a = 2$, while all invisible arcs carry no flow, i.e., $f_a = 0$. Figure 4.5b shows a solution featuring the maximum supply of six units. On the other hand, Figures 4.5c and 4.5d show solutions with a total supply of four units. Figure 4.5d shows an optimal solution.

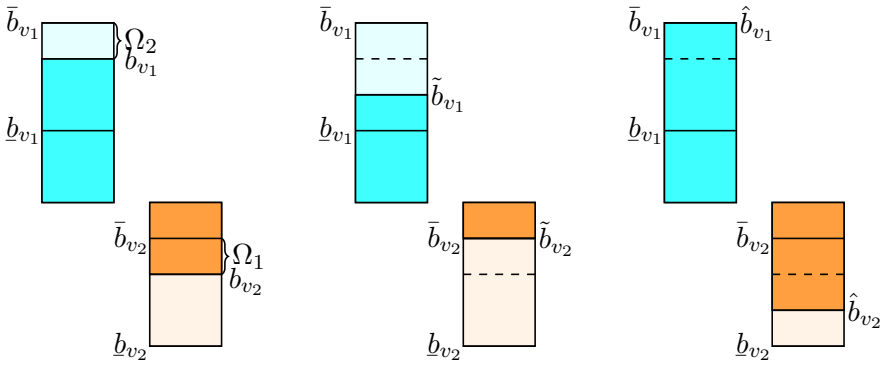


Figure 4.6.: Admissible supply and demand vector b with $v_1 \in \mathcal{V}^+$, $v_2 \in \mathcal{V}^-$, $b_{v_1} \in (\underline{b}_{v_1}, \bar{b}_{v_1})$, and $b_{v_2} \in (\underline{b}_{v_2}, \bar{b}_{v_2})$ (left). In $\tilde{b}$ (middle), the inflow and outflow have been decreased by Ω_1, while in $\hat{b}$ (right), both have been increased by Ω_2. In both of the latter supply and demand vectors, one node's demand or supply is equal to a bound, respectively.

Definition 1. *Let $\mathcal{I} = (\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$ be a MaxTP instance and let b be an admissible supply and demand vector. We call b bound-close if all of its values are equal to one of the corresponding bounds except for at most one, i.e., we have $b_v \in (\underline{b}_v, \bar{b}_v)$ for at most one node $v \in \mathcal{V}^+ \cup \mathcal{V}^-$.*

Lemma 1. *Let $\mathcal{I} = (\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$ be a MaxTP instance. There exists an optimal solution (b, f) with b being bound-close.*

Proof. Let (b, f) be an optimal solution for $\mathcal{I}$. If b is not bound-close, there exist two nodes $v_1, v_2 \in \mathcal{V}^+ \cup \mathcal{V}^-$ with $v_1 \neq v_2$ such that the corresponding supply or demand values are not equal to one of their bounds, i.e., we have $b_{v_1} \in (\underline{b}_{v_1}, \bar{b}_{v_1})$ and $b_{v_2} \in (\underline{b}_{v_2}, \bar{b}_{v_2})$.

Let $\Omega_1 := \min\{b_{v_1} - \underline{b}_{v_1}, \bar{b}_{v_2} - b_{v_2}\} > 0$ and $\Omega_2 := \min\{\bar{b}_{v_1} - b_{v_1}, b_{v_2} - \underline{b}_{v_2}\} > 0$. Consider the supply and demand vectors $\tilde{b}$ and $\hat{b}$ defined as

$$\tilde{b}_v := \begin{cases} b_{v_1} - \Omega_1 & \text{if } v = v_1 \\ b_{v_2} + \Omega_1 & \text{if } v = v_2 \\ b_v & \text{otherwise} \end{cases} \quad \text{and} \quad \hat{b}_v := \begin{cases} b_{v_1} + \Omega_2 & \text{if } v = v_1 \\ b_{v_2} - \Omega_2 & \text{if } v = v_2 \\ b_v & \text{otherwise.} \end{cases}$$

By construction, $\tilde{b}$ and $\hat{b}$ are admissible. Additionally, for $\tilde{b}$ it holds that $\tilde{b}_{v_1} = \underline{b}_{v_1}$ or $\tilde{b}_{v_2} = \bar{b}_{v_2}$, and for $\hat{b}$ we have that $\hat{b}_{v_1} = \bar{b}_{v_1}$ or $\hat{b}_{v_2} = \underline{b}_{v_2}$. An example illustrating these definitions is visualized in Figure 4.6.

Next, let $\tilde{f}$ and $\hat{f}$ denote optimal solutions for the TP instances induced by $\tilde{b}$ and $\hat{b}$, respectively. By construction $(\tilde{b}, \tilde{f})$ and $(\hat{b}, \hat{f})$ are feasible solutions for MaxTP, and we have $c(b, f) \geq c(\tilde{b}, \tilde{f})$ as well as $c(b, f) \geq c(\hat{b}, \hat{f})$ since (b, f) is optimal.

On the other hand, for $\lambda := \frac{\Omega_2}{\Omega_1 + \Omega_2} \in (0, 1)$, we have $\lambda\tilde{b} + (1 - \lambda)\hat{b} = b$. While this

trivially holds for all $v \in (\mathcal{V}^+ \cup \mathcal{V}^-) \setminus \{v_1, v_2\}$, for v_1, we have

$$\begin{aligned}
\lambda\,\tilde{b}_{v_1} + (1-\lambda)\,\hat{b}_{v_1} &= \frac{\Omega_2}{\Omega_1+\Omega_2}\,(b_{v_1} - \Omega_1) + \left(1 - \frac{\Omega_2}{\Omega_1+\Omega_2}\right)(b_{v_1} + \Omega_2) \\
&= \frac{\Omega_2}{\Omega_1+\Omega_2}\,(b_{v_1} - \Omega_1) + \frac{\Omega_1}{\Omega_1+\Omega_2}\,(b_{v_1} + \Omega_2) \\
&= \frac{b_{v_1}\Omega_2 - \Omega_1\Omega_2 + b_{v_1}\Omega_1 + \Omega_1\Omega_2}{\Omega_1+\Omega_2} \\
&= \frac{b_{v_1}(\Omega_1+\Omega_2)}{\Omega_1+\Omega_2} \;=\; b_{v_1},
\end{aligned}$$

and an analogous argument can be made for v_2. Since $\lambda \in (0,1)$, it also holds that $\lambda\tilde{f} + (1-\lambda)\hat{f}$ is a feasible flow for the TP instance induced by b since

$$(b, \lambda\tilde{f} + (1-\lambda)\hat{f}) \;=\; (\lambda\tilde{b} + (1-\lambda)\hat{b}, \lambda\tilde{f} + (1-\lambda)\hat{f}) \;=\; \lambda(\tilde{b},\tilde{f}) + (1-\lambda)(\hat{b},\hat{f}),$$

but it may not be optimal.

Nevertheless, we have $c(b,f) = c(\tilde{b},\tilde{f}) = c(\hat{b},\hat{f})$, i.e., it follows that the minimum transport moments are equal, because

$$\begin{aligned}
c(b,f) &\le c(b, \lambda\tilde{f} + (1-\lambda)\hat{f}) \\
&= \sum_{a\in\mathcal{A}} \ell_a(\lambda\tilde{f}_a + (1-\lambda)\hat{f}_a) \\
&= \lambda \sum_{a\in\mathcal{A}} \ell_a\tilde{f}_a + (1-\lambda)\sum_{a\in\mathcal{A}} \ell_a\hat{f}_a \\
&= \lambda\, c(\tilde{b},\tilde{f}) + (1-\lambda)\, c(\hat{b},\hat{f}) \\
&\le \lambda\, c(b,f) + (1-\lambda)\, c(b,f) \;=\; c(b,f).
\end{aligned}$$

Thus, $(\tilde{b},\tilde{f})$ and $(\hat{b},\hat{f})$ are optimal solutions with the supply or demand value of b_{v_1} or b_{v_2} being equal to one of the corresponding bounds. Iteratively applying this algorithmic procedure to node pairs $v_1, v_2 \in \mathcal{V}^+ \cup \mathcal{V}^-$ whose demand or supply values are not at their bounds, we derive an optimal solution with a bound-close supply and demand vector after at most $|\mathcal{V}^+ \cup \mathcal{V}^-| - 1$ iterations. □

4.2.2. A Flow-Direction Based Heuristic for MaxTP

Next, we present our Flow-Direction Based Min-Cost-Flow Method (FDB-MCF) for MaxTP. It is an adaption of the Reference Point Method (RPM) of Steringa et al. [153], which has been presented at the end of Subsection 4.1.2. Analogously, the basic idea behind FDB-MCF is to determine supply and demand vectors that are severe w.r.t. the minimum transport moment while covering all flow directions. Thereby, bound-closeness, which is the property of an optimal solution for MaxTP, see Lemma 1, is ensured in every step of the algorithm. Furthermore, in contrast to RPM, actual routings of the flow are considered within FDB-MCF.

Before stating FDB-MCF, we first introduce some additional notation. Given an admissible supply and demand vector b for MaxTP instance $(\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$, let us denote an optimal solution of the induced TP instance by $\mathrm{TP}(b)$. Furthermore, if we talk about distances between sources and sinks in this section, we consider the underlying undirected flow network when determining them. Based on this, we state FDB-MCF in Algorithm 5.

Algorithm 5: Flow-Direction Based Min-Cost-Flow Method (FDB-MCF)

Input : A MaxTP instance $(\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$
Output: A feasible and bound-close solution (b, f)

1 $c_{\max} \leftarrow 0$; $(b, f)_{\max} \leftarrow \emptyset$
2 **for** $\hat{u} \in \mathcal{V}^+$ **do**
3 $\quad (b, f)_{\hat{u}} \leftarrow \text{FDB-MCF}^+((\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b}), \hat{u})$
4 $\quad$ **if** $c((b, f)_{\hat{u}}) \geq c_{max}$ **then**
5 $\quad\quad c_{\max} \leftarrow c((b, f)_{\hat{u}})$; $(b, f)_{\max} \leftarrow (b, f)_{\hat{u}}$
6 **for** $\hat{w} \in \mathcal{V}^-$ **do**
7 $\quad (b, f)_{\hat{w}} \leftarrow \text{FDB-MCF}^-((\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b}), \hat{w})$
8 $\quad$ **if** $c((b, f)_{\hat{w}}) \geq c_{max}$ **then**
9 $\quad\quad c_{\max} \leftarrow c((b, f)_{\hat{w}})$; $(b, f)_{\max} \leftarrow (b, f)_{\hat{w}}$
10 **return** $(b, f)_{\max}$

FDB-MCF iterates over the entries and exits of the network and applies the corresponding subroutines FDB-MCF$^+$ and FDB-MCF$^-$. These subroutines then generate a feasible and bound-close MaxTP solution with the corresponding entry or exit representing the main direction of the flow into or out of the network, respectively. Finally, a solution with the maximum minimum transport moment among all the generated ones is returned.

Next, we explain FDB-MCF$^+$, which is stated in Algorithm 6 and applied to the entries of the network, in detail. Since FDB-MCF$^-$, which is stated in Algorithm 7, works analogously with the role of the entries and the exits being swapped, we refrain from a detailed discussion regarding it in the following.

FDB-MCF$^+$ starts with an admissible and bound-close supply and demand vector $b(\hat{u})$ in line 1, which is generated as follows: Consider the b-vector with $b_u = \underline{b}_u$ for all $u \in \mathcal{V}^+$ and $b_w = \bar{b}_w$ for all $w \in \mathcal{V}^-$. If $\sum_{v \in \mathcal{V}^+ \cup \mathcal{V}^-} b_v < 0$, we increase the supplies of the entries ordered non-decreasingly by their distance towards $\hat{u}$ up to their upper bounds until b is balanced. If $\sum_{v \in \mathcal{V}^+ \cup \mathcal{V}^-} b_v > 0$, we analogously increase the demands of the exits ordered non-increasingly by their distance towards $\hat{u}$ until b is balanced. Afterward, a first feasible solution $(b(\hat{u}), \mathrm{TP}(b(\hat{u})))$ is generated in line 2. Next, in each iteration of the main loop an entry $\tilde{u}$ whose supply is not at its upper bound and which is closest to $\hat{u}$ and an exit $\tilde{w}$ whose demand is not at its lower bound and which is farthest from $\hat{u}$ are chosen. Afterward, the supply and the demand of these two nodes are increased until one of them hits a bound, see lines 7–10. If the minimum transport moment of the new supply and

Algorithm 6: Subroutine FDB-MCF$^+$

Input : A MaxTP instance $(\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$, $\hat{u} \in \mathcal{V}^+$
Output: A feasible and bound-close solution (b, f)

1 $b(\hat{u}) \leftarrow$ Admissible and bound-close supply and demand vector w.r.t. $\hat{u}$
2 $(b, f)_{\hat{u}} \leftarrow (b(\hat{u}), \mathrm{TP}(b(\hat{u})))$; $(\tilde{b}, \tilde{f}) \leftarrow (b(\hat{u}), \mathrm{TP}(b(\hat{u})))$
3 **while** $c((b, f)_{\hat{u}}) \leq c(\tilde{b}, \tilde{f})$ **do**
4 $\quad (b, f)_{\hat{u}} \leftarrow (\tilde{b}, \tilde{f})$
5 $\quad$ **if** $b_u = \bar{b}_u$ *for all* $u \in \mathcal{V}^+$ *or* $b_w = \underline{b}_w$ *for all* $w \in \mathcal{V}^-$ **then**
6 $\quad\quad$ **return** $(b, f)_{\hat{u}}$
7 $\quad \tilde{u} \leftarrow$ Entry closest to $\hat{u}$ with $b_{\tilde{u}} < \bar{b}_{\tilde{u}}$
8 $\quad \tilde{w} \leftarrow$ Exit farthest from $\hat{u}$ with $|b_{\tilde{w}}| < |\underline{b}_{\tilde{w}}|$
9 $\quad \Omega \leftarrow \min\{\bar{b}_{\tilde{u}} - b_{\tilde{u}}, |\underline{b}_{\tilde{w}}| - |b_{\tilde{w}}|\}$
10 $\quad \tilde{b} \leftarrow b$; $\tilde{b}_{\tilde{u}} \leftarrow \tilde{b}_{\tilde{u}} + \Omega$; $\tilde{b}_{\tilde{w}} \leftarrow \tilde{b}_{\tilde{w}} - \Omega$
11 $\quad (\tilde{b}, \tilde{f}) \leftarrow (\tilde{b}, \mathrm{TP}(\tilde{b}))$
12 **return** $(b, f)_{\hat{u}}$

demand vector $\tilde{b}$ is greater than or equal to the minimum transport moment from the previous iteration, the algorithm continues with $(\tilde{b}, \tilde{f})$ as incumbent, see lines 3 and 4. Otherwise, the solution from the previous iteration is returned, see line 12. Additionally, the algorithm also terminates whenever supplies or demands cannot be increased any further, compare lines 5 and 6. Due to the construction of the initial solution $(b(\hat{u}), \mathrm{TP}(b(\hat{u})))$ and the choice of $\tilde{u}$ and $\tilde{w}$, the supply and demand vector b remains bound-close in every step of the algorithm, i.e., there is at most one entry or exit whose supply or demand is not equal to one of its bounds, respectively.

The role of $\hat{u}$ here is equivalent to the reference point in RPM: It describes the direction from where the flow is supposed to enter the network. However, by considering an optimal routing w.r.t. the TP instance induced by the incumbent supply and demand vector in each iteration, we avoid problems arising from cyclic structures. In particular, we rule out flow augmentations that might relax the overall transport situation and decrease the minimum transport moment.

We note that FDB-MCF returns the optimal solution shown in Figure 4.4 for the example MaxTP instance from Figure 4.2. To see this, consider FDB-MCF$^-$ applied to any of the exits.

4.2.3. Complexity of MaxTP

After introducing a first heuristic for MaxTP with FDB-MCF in the previous Subsection 4.2.2, the natural question arises whether there exists any polynomial-time algorithm for solving it or not. In particular, the underlying uncapacitated linear flow model may suggest a positive answer. However, in this section, we prove the following statement.

Algorithm 7: Subroutine FDB-MCF$^-$

Input : A MaxTP instance $(\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$, $\hat{w} \in \mathcal{V}^-$
Output: A feasible and bound-close solution (b, f)

$b(\hat{w}) \leftarrow$ Admissible and bound-close supply and demand vector w.r.t. $\hat{w}$
$(b, f)_{\hat{w}} \leftarrow (b(\hat{w}), \mathrm{TP}(b(\hat{w})))$; $(\tilde{b}, \tilde{f}) \leftarrow (b(\hat{w}), \mathrm{TP}(b(\hat{w})))$
while $c((b, f)_{\hat{w}}) \leq c(\tilde{b}, \tilde{f})$ **do**
  $(b, f)_{\hat{w}} \leftarrow (\tilde{b}, \tilde{f})$
  if $b_u = \bar{b}_u$ *for all* $u \in \mathcal{V}^+$ *or* $b_w = \underline{b}_w$ *for all* $w \in \mathcal{V}^-$ **then**
    return $(b, f)_{\hat{w}}$
  $\tilde{w} \leftarrow$ Exit closest to $\hat{w}$ with $b_{\tilde{u}} < \bar{b}_{\tilde{u}}$
  $\tilde{u} \leftarrow$ Entry farthest from $\hat{w}$ with $|b_{\tilde{w}}| < |\underline{b}_{\tilde{w}}|$
  $\Omega \leftarrow \min\{\bar{b}_{\tilde{u}} - b_{\tilde{u}}, |\underline{b}_{\tilde{w}}| - |b_{\tilde{w}}|\}$
  $\tilde{b} \leftarrow b$; $\tilde{b}_{\tilde{u}} \leftarrow \tilde{b}_{\tilde{u}} + \Omega$; $\tilde{b}_{\tilde{w}} \leftarrow \tilde{b}_{\tilde{w}} - \Omega$
  $(\tilde{b}, \tilde{f}) \leftarrow (\tilde{b}, \mathrm{TP}(\tilde{b}))$
return $(b, f)_{\hat{w}}$

Theorem 1. MaxTP *is NP-hard.*

We reduce from Partition. The following definition is adapted from SP12 in [55].

Definition 2. *Given a finite set* $Z := \{z_1, \ldots, z_n\}$ *and a size* $s(z) \in \mathbb{Z}^+$ *for each element* $z \in Z$*. Does there exist a feasible partition of* Z*, i.e., a set* $Z' \subseteq Z$ *such that* $\sum_{z \in Z'} s(z) = \sum_{z \in Z \setminus Z'} s(z)$*.*

Given an instance Z of Partition, we construct a corresponding MaxTP instance $\mathcal{I}_Z = (\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$ as follows. First of all, for each $z_i \in Z$, we add an entry $u_i \in \mathcal{V}^+$ and an exit $w_i \in \mathcal{V}^-$. For each entry $u_i \in \mathcal{V}^+$, we set $\underline{b}_{u_i} := 0$ and $\bar{b}_{u_i} := s(z_i)$, while for each exit $w_i \in \mathcal{V}^-$, we set $\underline{b}_{w_i} = -s(z_i)$ and $\bar{b}_{w_i} := 0$. Additionally, we add a single inner node $v \in \mathcal{V}^0$.

The set $\mathcal{A}$ consists of three different arc types, i.e., $\mathcal{A} := \mathcal{A}^1 \cup \mathcal{A}^2 \cup \mathcal{A}^3$. First, for each entry $u_i \in \mathcal{V}^+$, an arc towards the corresponding exit $w_i \in \mathcal{V}^-$ is added, i.e., $\mathcal{A}^1 := \{(u_i, w_i) \mid i \in \{1, \ldots, n\}\}$, and we set $\ell_a := 0$ for each $a \in \mathcal{A}^1$. Second, an arc between each entry $u_i \in \mathcal{V}^+$ and v is added, i.e., $\mathcal{A}^2 := \{(u_i, v) \mid i \in \{1, \ldots, n\}\}$, and we set $\ell_a := 1$ for each $a \in \mathcal{A}^2$. Third, an arc between the inner node v and each exit $w_i \in \mathcal{V}^-$ is added, i.e., $\mathcal{A}^3 := \{(v, w_i) \mid i \in \{1, \ldots, n\}\}$, and we set $\ell_a := 1$ for each $a \in \mathcal{A}^3$. This concludes the construction of instance $\mathcal{I}_Z$, which features $|\mathcal{V}| = 2|Z| + 1$ vertices and $|\mathcal{A}| = 3|Z|$ arcs. Figure 4.7 shows the MaxTP instance $\mathcal{I}_Z$ corresponding to the example Partition instance described in its caption.

Next, let us call a supply and demand vector b for $\mathcal{I}_Z$ complementary if $b_{u_i} = 0$ or $b_{w_i} = 0$ holds for all $i \in \{1, \ldots, n\}$.

Lemma 2. *For each feasible solution* (b, f) *of* $\mathcal{I}_Z$*, there exists a feasible solution* $(\tilde{b}, \tilde{f})$ *with complementary* $\tilde{b}$ *such that* $c(b, f) = c(\tilde{b}, \tilde{f})$*.*

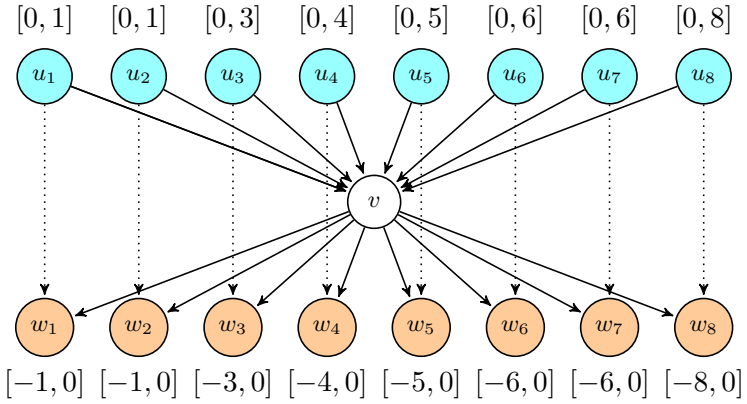


Figure 4.7.: MaxTP instance $\mathcal{I}_Z$ for the Partition instance $Z := \{z_1, \dots z_8\}$ with $s(z_1) = 1$, $s(z_2) = 1$, $s(z_3) = 3$, $s(z_4) = 4$, $s(z_5) = 5$, $s(z_6) = 6$, $s(z_7) = 6$, and $s(z_8) = 8$. The dotted arcs represent $\mathcal{A}^1$ and have length $\ell_a = 0$, while the solid arcs represent $\mathcal{A}^2$ and $\mathcal{A}^3$ and have length $\ell_a = 1$. The supply and demand intervals are stated above and below the entries and exits, respectively.

Proof. For each $i \in \{1, \dots, n\}$ let $\Omega_i := \min\{b_{v_i}, |b_{w_i}|\} \geq 0$. Since f is an optimal solution for the TP instance induced by b, we have $f_{u_i w_i} = \Omega_i$. Thus,

$$\tilde{b}_v := \begin{cases} b_v - \Omega_i & \text{if } v = u_i \in \mathcal{V}^+ \\ b_v + \Omega_i & \text{if } v = w_i \in \mathcal{V}^- \end{cases} \text{ and } \tilde{f}_a := \begin{cases} f_a - \Omega_i & \text{if } a = (u_i, w_i) \in \mathcal{A}^1 \\ f_a & \text{otherwise} \end{cases}$$

is a feasible solution with complementary $\tilde{b}$. Additionally, since $\ell_a = 0$ for all $a \in \mathcal{A}^1$, we have $c(b, f) = c(\tilde{b}, \tilde{f})$. □

Corollary 1. *There exists an optimal solution (b, f) for $\mathcal{I}_Z$ with complementary supply and demand vector b.*

Lemma 3. *There exists a feasible partition Z' of Z if and only if there exists a feasible solution (b, f) for $\mathcal{I}_Z$ with $c(b, f) \geq \sum_{z \in Z} s(z)$.*

Proof. Let $Z' \subseteq Z$ be a feasible partition. Therefore, it holds by definition that $\sum_{z \in Z'} s(z) = \sum_{z \in Z} \frac{s(z)}{2} = \sum_{z \in Z \setminus Z'} s(z)$. Consider (b, f) defined as

$$b_v := \begin{cases} s(z_i) & \text{if } v = u_i \in \mathcal{V}^+,\ z_i \in Z' \\ -s(z_i) & \text{if } v = w_i \in \mathcal{V}^-,\ z_i \in Z \setminus Z' \\ 0 & \text{otherwise,} \end{cases}$$

and

$$f_a := \begin{cases} s(z_i) & \text{if } a = (u_i, v) \in \mathcal{A}^2,\ z_i \in Z' \\ |s(z_i)| & \text{if } a = (v, w_i) \in \mathcal{A}^3,\ z_i \in Z \setminus Z' \\ 0 & \text{otherwise.} \end{cases}$$

By construction, b is an admissible supply and demand vector and complementary, f represents an optimal solution for the TP instance induced by b, and we have

$$\begin{aligned} c(b, f) &= \sum_{(u_i,v)\in\mathcal{A}^2} f_{u_i v} + \sum_{(v,w_i)\in\mathcal{A}^3} f_{v w_i} = \sum_{z_i\in Z'} f_{u_i v} + \sum_{z_i\in Z\setminus Z'} f_{v w_i} \\ &= \sum_{z_i\in Z'} b_{u_i} + \sum_{z_i\in Z\setminus Z'} |b_{w_i}| = \sum_{z_i\in Z'} s(z_i) + \sum_{z_i\in Z\setminus Z'} s(z_i) \\ &= \sum_{z_i\in Z} s(z_i). \end{aligned}$$

Conversely, by Lemma 2 there exists a feasible and complementary solution (b, f) for $\mathcal{I}_Z$ such that $c(b, f) \geq \sum_{z\in Z} s(z)$. Due to the complementarity of b, we have $f_a = 0$ for all $a \in \mathcal{A}^1$. Let $Z' := \{z_i \in Z \mid b_{u_i} > 0\} \subseteq Z$, then

$$\begin{aligned} \sum_{z\in Z} s(z) \leq c(b, f) &= \sum_{(u_i,v)\in\mathcal{A}^2} f_{u_i v} + \sum_{(v,w_i)\in\mathcal{A}^3} f_{v w_i} \\ &= \sum_{(u_i,v)\in\mathcal{A}^2} b_{u_i} + \sum_{(v,w_i)\in\mathcal{A}^3} |b_{w_i}| \\ &= \sum_{z_i\in Z'} b_{u_i} + \sum_{z_i\in Z\setminus Z'} |b_{w_i}| \\ &\leq \sum_{z_i\in Z'} s(z_i) + \sum_{z_i\in Z\setminus Z'} s(z_i) = \sum_{z\in Z} s(z). \end{aligned}$$

This shows that $c(b, f) = \sum_{z\in Z} s(z)$ as well as $\sum_{z_i\in Z'} b_{u_i} = \sum_{z_i\in Z'} s(z_i)$ and $\sum_{z_i\in Z\setminus Z'} |b_{w_i}| = \sum_{z_i\in Z\setminus Z'} s(z_i)$ hold with equality. Furthermore, since b is balanced, it follows that

$$\sum_{z_i\in Z'} b_{u_i} = \sum_{z_i\in Z'} s(z_i) = \sum_{z\in Z} \frac{s(z)}{2} = \sum_{z_i\in Z\setminus Z'} s(z_i) = \sum_{z_i\in Z\setminus Z'} |b_{w_i}|,$$

showing that Z' is a feasible partition. □

Theorem 1. MaxTP *is NP-hard.*

Proof. Deciding whether there exists a feasible partition or not is an NP-complete problem, see SP12 in [55]. Hence, MaxTP is NP-hard, since any polynomial-time algorithm applied to $\mathcal{I}_Z$, deciding whether it admits a feasible solution (b, f) with $c(b, f) \geq \sum_{z\in Z} s(z)$ or not, could also be used to decide whether Z admits a feasible partition by Lemma 3 or not. □

4.2.4. Bilevel Programming Model

Bilevel programming is often applied to model problems related to energy transport, see the book chapter of Wogrin et al. [162] and the references therein. Hence, it is not surprising that MaxTP can be modeled as a linear bilevel program with interdicting objective functions.

$$\max_{b} \quad \sum_{a\in\mathcal{A}} \ell_a f_a \tag{4.1}$$

$$\text{s.t.} \quad \sum_{u\in\mathcal{V}^+} b_u + \sum_{w\in\mathcal{V}^-} b_w = 0 \tag{4.2}$$

$$b_v \in [\underline{b}_v, \bar{b}_v] \qquad \forall v \in \mathcal{V}^+ \cup \mathcal{V}^- \tag{4.3}$$

$$\min_{f} \quad \sum_{a\in\mathcal{A}} \ell_a f_a \tag{4.4}$$

$$\text{s.t.} \sum_{a\in\delta^+(v)} f_a - \sum_{a\in\delta^-(v)} f_a = b_v \qquad \forall v \in \mathcal{V}^+ \cup \mathcal{V}^- \tag{4.5}$$

$$\sum_{a\in\delta^+(v)} f_a - \sum_{a\in\delta^-(v)} f_a = 0 \qquad \forall v \in \mathcal{V}^0 \tag{4.6}$$

$$f_a \in \mathbb{R}_{\geq 0} \qquad \forall a \in \mathcal{A} \tag{4.7}$$

For each entry or exit $v \in \mathcal{V}^+ \cup \mathcal{V}^-$, the variable b_v represents its supply or demand, respectively. These values are chosen by the leader obeying the corresponding bounds, see (4.3). Furthermore, we note that in analogy to the natural gas market, the leader has to balance supply and demand. Otherwise the induced linear programming model (4.4)–(4.7) for TP, which the follower solves subsequently, does not admit a feasible solution. However, to emphasize this necessity, we add the redundant constraint (4.2). Hence, the resulting b is an admissible supply and demand vector. While the follower routes the flow such that the cost is minimized, the leader's goal is to maximize the resulting minimum transport moment, compare (4.4) and (4.1), respectively.

MaxTP can be considered as some variant of a network interdiction problems. This class of problems is for example discussed by Smith and Lim [150], Smith and Song [151], and Wood [163]. For instance, in the Minimum Cost Flow Interdiction Problem, the supplies and demands of the entries and exits are fixed. However, the leader has a budget to decrease the arc capacities and thereby tries to maximize the optimal objective value of the MCF problem of the follower [150].

4.2.5. Classical KKT Reformulation

A common strategy to solve linear bilevel optimization formulations, such as the one for MaxTP introduced in the previous Subsection 4.2.4, is to reduce them to single-level problems. We apply a classical KKT reformulation here. Thereby, the optimization problem of the follower is replaced by its KKT conditions, a set of

necessary and sufficient optimality conditions, see Section 2.2 for more details. In our particular case, we remove the lower level and add the constraints and variables of the corresponding primal and dual linear program for TP and the KKT complementarity conditions to the upper level. For our model (4.1)–(4.7), the resulting nonlinear program reads

$$\max_{b,f,\phi,\pi} \quad \sum_{a\in\mathcal{A}} \ell_a f_a \tag{4.8}$$

$$\text{s.t.} \quad \sum_{u\in\mathcal{V}^+} b_u + \sum_{w\in\mathcal{V}^-} b_w = 0 \tag{4.9}$$

$$\sum_{a\in\delta^+(v)} f_a - \sum_{a\in\delta^-(v)} f_a = b_v \qquad \forall v \in \mathcal{V}^+ \cup \mathcal{V}^- \tag{4.10}$$

$$\sum_{a\in\delta^+(v)} f_a - \sum_{a\in\delta^-(v)} f_a = 0 \qquad \forall v \in \mathcal{V}^0 \tag{4.11}$$

$$\pi_v - \pi_u + \phi_a = \ell_a \qquad \forall (u,v) = a \in \mathcal{A} \tag{4.12}$$

$$\phi_a f_a = 0 \qquad \forall a \in \mathcal{A} \tag{4.13}$$

$$b_v \in [\underline{b}_v, \bar{b}_v] \qquad \forall v \in \mathcal{V}^+ \cup \mathcal{V}^- \tag{4.14}$$

$$f_a \in \mathbb{R}_{\geq 0} \qquad \forall a \in \mathcal{A} \tag{4.15}$$

$$\phi_a \in \mathbb{R}_{\geq 0} \qquad \forall a \in \mathcal{A} \tag{4.16}$$

$$\pi_v \in \mathbb{R} \qquad \forall v \in \mathcal{V}. \tag{4.17}$$

Regarding the dual linear program, the π-variables correspond to the flow conservation constraints (4.10) and (4.11), while the ϕ_a-variables correspond to the nonnegativity constraints of the flow variables (4.15). Furthermore, the dual constraints are stated in (4.12). On the other hand, the KKT complementarity conditions can be found in (4.13).

The additional variables and constraints in this nonlinear model can be interpreted as follows. The π-variables can be seen as node potentials. If there is nonzero flow on some arc $a = (u,v) \in \mathcal{A}$ in a solution, then the potential difference between π_v and π_u must be equal to its length ℓ_a. This ensures, that there exists no cycle of nonzero length with nonzero flow on it, which is a well-known property of optimal minimum cost flows. On the other hand, if there is no flow on a, then the potential difference has to be greater than or equal to ℓ_a, and equality in constraints (4.12) is ensured by the ϕ_a-variables.

One way to directly solve nonlinear mathematical programs resulting from KKT reformulations of linear bilevel optimization problems is to apply the SOS-1 technique. Thereby, the complementarity conditions are initially omitted and then branched on later on during the solving process. For more details on this method, we refer to the work of Fortuny-Amat and McCarl [47] and Bard and Moore [9].

The second, very popular method is to apply a big-M reformulation of the complementarity conditions, which was first described by Fortuny-Amat and McCarl [47]. Here, an auxiliary binary variable is introduced for each complementarity condition,

which is then replaced by two linear constraints. Depending on the value of the corresponding binary variable, these linear constraints force one of the two terms to be equal to zero. However, valid upper bounds are needed to derive a correct MILP model, as demonstrated and discussed by Pineda and Morales [132].

Computational experiments conducted by Kleinert and Schmidt [95] show that the big-M approach should be preferred if valid (and small) bounds are available. Hence, the overall goal of the following three subsections is to determine (small) upper bounds on the ϕ- and f-variables and to eventually obtain a valid MILP model for MaxTP. Since verifying the correctness of upper bounds is itself an NP-hard problem, see Kleinert et al. [94], we will exploit problem-specific knowledge and the structure of the underlying flow network to derive them.

4.2.6. Solution-Equivalency and L^1 Instances

This subsection introduces an equivalence relation for MaxTP instances based on the distances between the entries and the exits of the underlying flow networks. In particular, we will call two MaxTP instances solution-equivalent if their feasible solutions can easily be mapped onto each other while the objective function values are preserved. Moreover, we derive a method to generically derive solution-equivalent MaxTP instances whose flow networks are acyclic, tripartite, and small w.r.t. the number of arcs. For these L^1 instances, whose name stands for the one additional layer of nodes between the entries and exits, we can prove variable bounds for the corresponding KKT reformulation from Subsection 4.2.7. Moreover, these bounds are then used to define a MIP model in Subsection 4.2.8.

Hence, our motivation here can be summarized as follows: The idea for solving an arbitrary MaxTP instance $\mathcal{I}$ is to generate a solution-equivalent L^1 instance, derive variable bounds for its KKT reformulation, solve the resulting MIP model, and map the obtained solution back to $\mathcal{I}$.

Recall that the distance $d_{v_1v_2}$ between two nodes $v_1, v_2 \in \mathcal{V}$ is defined as the length of a shortest path between them. Further, we note that $d_{uw} < \infty$ holds for all $u \in \mathcal{V}^+$ and $w \in \mathcal{V}^-$ due to the connectedness-condition, i.e., we assume that there always exists a direct path between them.

Definition 3. *Let $\mathcal{I} = (\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$ and $\tilde{\mathcal{I}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{A}}, \tilde{\ell}, \tilde{\underline{b}}, \tilde{\bar{b}})$ be MaxTP instances. We call $\mathcal{I}$ and $\tilde{\mathcal{I}}$ solution-equivalent if there exists a bijection $g : \mathcal{V}^+ \cup \mathcal{V}^- \to \tilde{\mathcal{V}}^+ \cup \tilde{\mathcal{V}}^-$ such that $d_{uw} = d_{g(u)g(w)}$ for all $u \in \mathcal{V}^+$ and $w \in \mathcal{V}^-$ as well as $\underline{b}_v = \tilde{\underline{b}}_{g(v)}$ and $\bar{b}_v = \tilde{\bar{b}}_{g(v)}$ for all $v \in \mathcal{V}^+ \cup \mathcal{V}^-$.*

Clearly, solution-equivalency constitutes an equivalence relation for MaxTP instances. Next, we give an example of a first generic solution-equivalent instance.

Definition 4. *Let $\mathcal{I} = (\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$ be a MaxTP instance. The MaxTP instance $\mathcal{I}^{\mathcal{P}} = (\mathcal{V}^{\mathcal{P}}, \mathcal{A}^{\mathcal{P}}, \ell^{\mathcal{P}}, \underline{b}, \bar{b})$ defined as $\mathcal{V}^{\mathcal{P}} := \mathcal{V}^+ \cup \mathcal{V}^-$, $\mathcal{A}^{\mathcal{P}} := \{(u, w) \mid u \in \mathcal{V}^+, w \in \mathcal{V}^-\}$, and $\ell^{\mathcal{P}}_{uw} := d_{uw}$ is called the path instance of $\mathcal{I}$.*

Corollary 2. *Let $\mathcal{I}$ be a MaxTP instance. Then $\mathcal{I}^{\mathcal{P}}$ is solution-equivalent.*

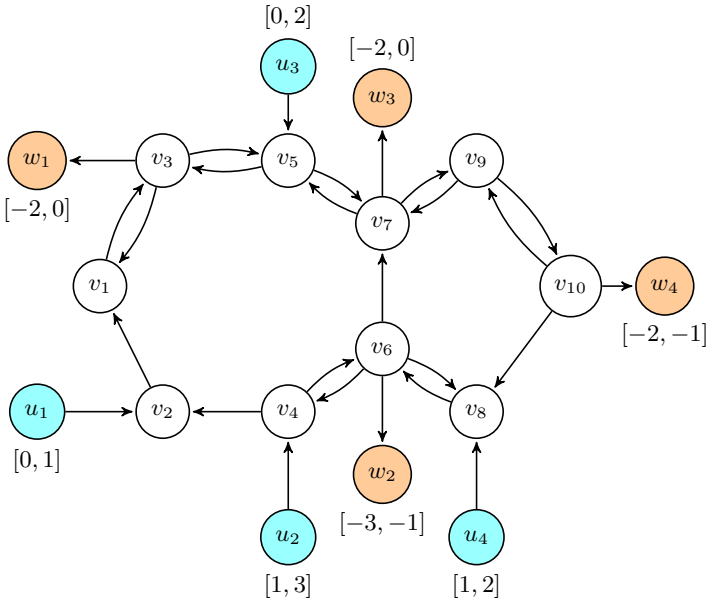


Figure 4.8.: Example MaxTP instance $\mathcal{I}$ with entries $\mathcal{V}^+ = \{u_1, u_2, u_3, u_4\}$ and exits $\mathcal{V}^- = \{w_1, w_2, w_3, w_4\}$. The supply and demand intervals are shown above or below the corresponding nodes. Further, we consider $\ell_a = 1$ for each $a \in \mathcal{A}$ as its length.

Proof. Using the identity function as bijection yields the desired result. □

The path instance $\mathcal{I}^{\mathcal{P}}$ corresponding to the example instance $\mathcal{I}$ from Figure 4.8 is shown in Figure 4.9. With its definition, we have simultaneously introduced a first generic way to derive a solution-equivalent MaxTP instance.

Next, we show how feasible solutions can be mapped between solution-equivalent instances. Therefore, we introduce the notion of a shortest path system.

Definition 5. *Let $\mathcal{I} = (\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$ be a MaxTP instance and let p_{uw} denote a shortest path from $u \in \mathcal{V}^+$ to $w \in \mathcal{V}^-$. Then $\mathcal{S} := \bigcup_{u \in \mathcal{V}^+} \bigcup_{w \in \mathcal{V}^-} p_{uw}$ is called a shortest path system (SPS) for $\mathcal{I}$.*

Again, since there exists a path between every $u \in \mathcal{V}^+$ and every $w \in \mathcal{V}^-$ due to the connectedness-condition, there always exists an SPS for $\mathcal{I}$.

Lemma 4. *Let $\mathcal{I} = (\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$ and $\tilde{\mathcal{I}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{A}}, \tilde{\ell}, \tilde{\underline{b}}, \tilde{\bar{b}})$ be solution-equivalent MaxTP instances with bijection g. Further, let (b, f) be a feasible solution for $\mathcal{I}$. Then there exists a feasible solution $(\tilde{b}, \tilde{f})$ for $\tilde{\mathcal{I}}$ with $\tilde{b} = g(b)$ and $c(b, f) = c(\tilde{b}, \tilde{f})$.*

Proof. First, we partition the arc flows f into flows on shortest paths between the entries and the exits of the network. Therefore, consider the subgraph $G' \subseteq G$ induced by the arc set $\{a \in \mathcal{A} \mid f_a > 0\}$. Any path between an entry and an

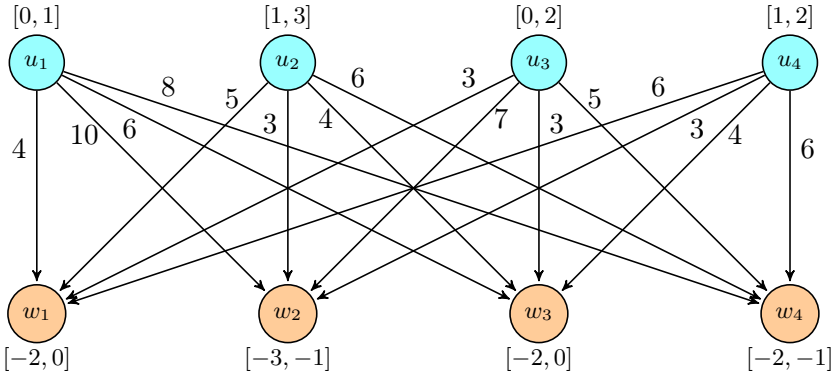


Figure 4.9.: Path instance $\mathcal{I}^{\mathcal{P}}$ for the example instance $\mathcal{I}$ from Figure 4.8. The arc lengths, i.e., the lengths of shortest paths between the entries and exits in $\mathcal{I}$, are depicted next to the corresponding entities.

exit in G' is a shortest path between them in G. Otherwise, there would exist a negative cycle in the residual graph of G contradicting the optimality of f for the induced TP instance. Thus, by consecutively determining these shortest paths and decreasing the corresponding supplies, demands, and flows, we partition f into flows $f_{p_{uw}}$ between the entries and exits. Next, given any SPS $\tilde{\mathcal{S}}$ for $\tilde{\mathcal{I}}$

$$\tilde{b}_v := b_{g^{-1}(v)} \text{ for } v \in \tilde{\mathcal{V}}^+ \cup \tilde{\mathcal{V}}^- \text{ and } \tilde{f}_a := \sum_{\substack{p_{uw} \in \tilde{\mathcal{S}} \\ a \in p_{uw}}} f_{p_{g^{-1}(u) g^{-1}(w)}} \text{ for } a \in \tilde{\mathcal{A}}$$

is a feasible solution for $\tilde{\mathcal{I}}$ with $c(b, f) = c(\tilde{b}, \tilde{f})$. □

Corollary 3. *Consider the setup from Lemma 4. Let (b, f) be an optimal solution for $\mathcal{I}$. There exists an optimal solution $(\tilde{b}, \tilde{f})$ for $\tilde{\mathcal{I}}$ with $\tilde{b} = g(b)$ and $c(b, f) = c(\tilde{b}, \tilde{f})$.*

Corollary 4. *Let $\mathcal{I} = (\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$ be a MaxTP instance, (b, f) be a feasible solution for $\mathcal{I}$, and $\mathcal{S}$ be a shortest path system for $\mathcal{I}$. There exists a feasible solution $(\tilde{b}, \tilde{f})$ for $\mathcal{I}$ where the arc flows can be partitioned into flows on the shortest paths in $\mathcal{S}$.*

Proof. Consider the corresponding path instance $\mathcal{I}^{\mathcal{P}}$. Its unique shortest path system $\tilde{\mathcal{S}}$ consists of the direct arcs between the entries and exits. Using the identity function as bijection g and applying the algorithmic procedure described in Lemma 4 in forward and backward direction, yields the desired result. □

Next, we introduce a procedure for deriving another solution-equivalent MaxTP instance, which is shall be small w.r.t. the number of arcs. Therefore, we first observe that the path instance $\mathcal{I}^{\mathcal{P}}$ from Definition 4 is a solution-equivalent MaxTP instance with $|\mathcal{V}^{\mathcal{P}}| = |\mathcal{V}^+| + |\mathcal{V}^-|$ nodes and $|\mathcal{A}^{\mathcal{P}}| = |\mathcal{V}^+| \cdot |\mathcal{V}^-|$ arcs. While the number of nodes is minimal w.r.t. our equivalency relation, the number of arcs can

become quite large. However, adding an additional intermediate node can reduce the number of arcs significantly, as the following example demonstrates.

Example 1. *Let $\mathcal{I}$ be an MaxTP instance. Assume there exists an SPS $\mathcal{S}$ and some $v \in \mathcal{V}^0$ such that $v \in p_{uw}$ for all $p_{uw} \in S$. Then the instance $\tilde{\mathcal{I}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{A}}, \tilde{\ell}, \underline{b}, \bar{b})$ defined as $\tilde{\mathcal{V}} := \mathcal{V}^+ \cup \mathcal{V}^- \cup \{v\}$, $\tilde{\mathcal{A}} := \{(u, v) \,|\, u \in \mathcal{V}^+\} \cup \{(v, w) \,|\, w \in \mathcal{V}^-\}$ as well as $\ell_{uv} := d_{uv}$ and $\ell_{vw} := d_{vw}$ is solution-equivalent with the identity function as bijection. Further, we have $|\tilde{\mathcal{V}}| := |\mathcal{V}^+| + |\mathcal{V}^-| + 1$ and $|\tilde{\mathcal{A}}| := |\mathcal{V}^+| + |\mathcal{V}^-|$.*

Inspired by this example and given some SPS $\mathcal{S}$, the idea for deriving a whole class of generic solution-equivalent MaxTP instances is to identify subsets of inner nodes $\mathcal{L} \subseteq \mathcal{V}^0$, which we call layer nodes in the following, such that many shortest paths in $\mathcal{S}$ contain at least one node in $\mathcal{L}$. Based on this, we define a tripartite flow network consisting of the entries, the exits, and the layer nodes, which gives rise to the solution-equivalent L^1 instances.

Definition 6. *Let $\mathcal{I}$ be a MaxTP instance and let $\mathcal{S}$ be an SPS for it. Furthermore, let $\mathcal{L} \subseteq \mathcal{V}^0$ and $d : \mathcal{S} \to \mathcal{L} \cup \{0\}$ be a map from the paths contained in $\mathcal{S}$ towards $\mathcal{L}$ together with an auxiliary value 0 such that $d(p_{uw}) = v \in \mathcal{L}$ only if $v \in p_{uw}$. Then $\mathcal{I}^{\mathcal{S},\mathcal{L},d} := (\tilde{\mathcal{V}}, \tilde{\mathcal{A}}, \tilde{\ell}, \underline{b}, \bar{b})$ defined as*

$$\begin{aligned}
\tilde{\mathcal{V}} &:= \mathcal{V}^+ \cup \mathcal{V}^- \cup \mathcal{L} \\
\tilde{\mathcal{A}} &:= \{(u, v), (v, w) \,|\, \text{for } u \in \mathcal{V}^+, w \in \mathcal{V}^-, \text{ and } v \in \mathcal{L} \text{ with } d(p_{uw}) = v\} \cup \\
&\qquad \{(u, w) \,|\, \text{for } u \in \mathcal{V}^+ \text{ and } w \in \mathcal{V}^- \text{ with } d(p_{uw}) = 0\}
\end{aligned}$$

with $\ell_{uv} := d_{uv}$, $\ell_{vw} := d_{vw}$, and $\ell_{uw} := d_{uw}$ is called an L^1 instance for $\mathcal{I}$.

Corollary 5. *Let $\mathcal{I}$ be a MaxTP instance together with an SPS $\mathcal{S}$, $\mathcal{L} \subseteq \mathcal{V}^0$, and a mapping d as defined in Definition 6. Then $\mathcal{I}$ and $\mathcal{I}^{\mathcal{S},\mathcal{L},d}$ are solution-equivalent.*

Corollary 6. *Let $\mathcal{I}$ be a MaxTP instance together with an SPS $\mathcal{S}$ and the mapping d with $d(p_{uw}) = 0$ for all $p_{uw} \in \mathcal{S}$. Then $\mathcal{I}^{\mathcal{P}} = \mathcal{I}^{\mathcal{S},\emptyset,d}$.*

An example L^1 instance for the MaxTP instance from Figure 4.8 is shown in Figure 4.10.

As mentioned at the beginning of this section, one of the goals is to obtain a solution-equivalent instance having a small number of arcs. Thus, given an instance $\mathcal{I}$ of MaxTP and some SPS $\mathcal{S}$ for it, the L^1 instance for $\mathcal{I}$ with the minimum number of arcs can be determined through solving the following set cover problem, which we model as a binary program.

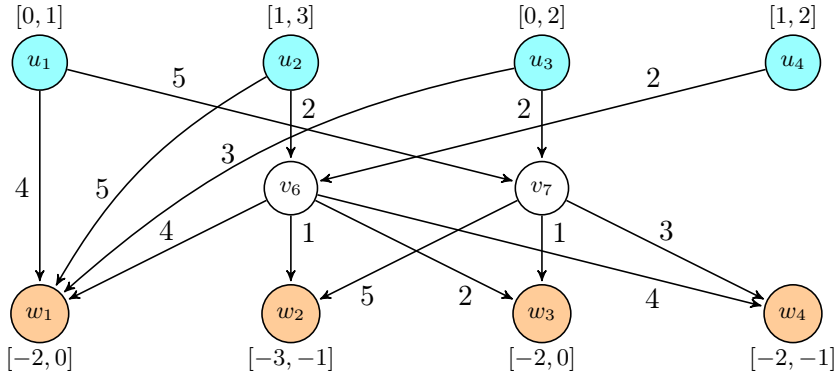


Figure 4.10.: Visualization of the L^1 instance $\mathcal{I}^{\mathcal{S},\mathcal{L},d}$ for example MaxTP instance $\mathcal{I}$ from Figure 4.8 defined with some arbitrary $\mathcal{S}$, $\mathcal{L} := \{v_6, v_7\}$ and the mapping d given as $d(p_{u_1w_1}) = 0$, $d(p_{u_1w_2}) = v_7$, $d(p_{u_1w_3}) = v_7$, $d(p_{u_1w_4}) = v_7$, $d(p_{u_2w_1}) = 0$, $d(p_{u_2w_2}) = v_6$, $d(p_{u_2w_3}) = v_6$, $d(p_{u_2w_4}) = v_6$, $d(p_{u_3w_1}) = 0$, $d(p_{u_3w_2}) = v_7$, $d(p_{u_3w_3}) = v_7$, $d(p_{u_3w_4}) = v_7$, $d(p_{u_4w_1}) = v_6$, $d(p_{u_4w_2}) = v_6$, $d(p_{u_4w_3}) = v_6$, $d(p_{u_4w_4}) = v_6$. The lengths can be found next to the corresponding arcs.

$$\min \sum_{u\in\mathcal{V}^+}\sum_{w\in\mathcal{V}^-} x_{uw} + \sum_{v\in V^0}\Big(\sum_{u\in\mathcal{V}^+} x_{uv} + \sum_{w\in\mathcal{V}^-} x_{vw}\Big) \tag{4.18}$$

$$\text{s.t.} \quad 2d^v_{uw} - x_{uv} - x_{vw} \le 0 \qquad \forall p_{uw} \in \mathcal{S},\, \forall v \in p_{uw} \cap \mathcal{V}^0 \tag{4.19}$$

$$d^0_{uw} - x_{uw} = 0 \qquad \forall p_{uw} \in \mathcal{S} \tag{4.20}$$

$$d^0_{uw} + \sum\nolimits_{v\in p_{uw}\cap\mathcal{V}^0} d^v_{uw} = 1 \qquad \forall p_{uw} \in \mathcal{S} \tag{4.21}$$

$$x_{uv}, x_{vw}, x_{vw}, d^v_{uw}, d^0_{uw} \in \{0,1\}. \tag{4.22}$$

The binary variables d^v_{uw} and d^0_{uw} encode the mapping $d : \mathcal{S} \to \mathcal{L} \cup \{0\}$. In particular, d^v_{uw} indicates whether $d(p_{uw}) = v$ for $v \in p_{uw} \cap \mathcal{V}^0$ or not. Additionally, due to constraint (4.21), each path $p_{uw} \in \mathcal{S}$ is assigned to at most one of its inner nodes. On the other hand, the binary variables x_{uv}, x_{vw}, and x_{uw} indicate whether the corresponding arcs must be added or not. Thereby, constraint (4.19) ensures that (u, v) and (v, w) are added if $d^v_{uw} = 1$,while constraint (4.20) ensures that (u, w) is added in case that $d^0_{uw} = 1$. The smallest set $\mathcal{L} \subseteq \mathcal{V}^0$ necessary to define d can then be retrieved from the solution.

However, besides being NP-hard in general, see SP5 in Garey and Johnson [55], set cover problems are often intrinsically hard to solve in practice. Therefore, we introduce a greedy heuristic for the mathematical program above to derive a solution-equivalent L^1 instance with a small number of arcs in Algorithm 8. The heuristic starts with the path instance $\mathcal{I}^{\mathcal{P}}$ as initial L^1 instance, see lines 2–4. Afterward, the

idea is to iteratively identify inner nodes in the original underlying flow network, whose addition to $\mathcal{L}$ reduces the number of arcs w.r.t. the incumbent L^1 instance. Therefore, we greedily choose a node $v \in \mathcal{V}^0 \setminus \mathcal{L}$ that leads to a biggest relative reduction r_v, i.e., whose addition maximizes the ratio between number of direct arcs between the entries and the exits that would be removed, which is equal to $|\mathcal{S}_v|$, and the number of new arcs incident to v that would have to be added, i.e., $|\mathcal{V}_v^+| + |\mathcal{V}_v^-|$, see lines 10 and 22. The algorithm stops as soon as the maximum relative ratio among the remaining inner nodes that have not been added becomes smaller than or equal to one, see lines 13 and 23.

Algorithm 8: Greedy heuristic for (4.18)–(4.22) to determine a solution-equivalent L^1 instance with a small number of arcs

Input : A MaxTP instance $\mathcal{I} = (\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$
Output: A solution-equivalent L^1 instance $\mathcal{I}^{\mathcal{S},\mathcal{L},d}$

1 $\mathcal{S} \leftarrow$ Generate an SPS for $\mathcal{I}$
2 $\mathcal{L} \leftarrow \emptyset$
3 **for** $p_{uw} \in \mathcal{S}$ **do**
4 $\quad d(p_{uw}) \leftarrow 0$
5
6 **for** $v \in \mathcal{V}^0$ **do**
7 $\quad \mathcal{S}_v \leftarrow \{p_{uw} \in \mathcal{S} \mid v \in p_{uw}\}$
8 $\quad \mathcal{V}_v^+ \leftarrow \{u \in \mathcal{V}^+ \mid \exists w \in \mathcal{V}^- \text{ such that } p_{uw} \in \mathcal{S}_v\}$
9 $\quad \mathcal{V}_v^- \leftarrow \{w \in \mathcal{V}^- \mid \exists u \in \mathcal{V}^+ \text{ such that } p_{uw} \in \mathcal{S}_v\}$
10 $\quad r_v \leftarrow \frac{|\mathcal{S}_v|}{\max\{1, |\mathcal{V}_v^+| + |\mathcal{V}_v^-|\}}$
11 $\tilde{v} \leftarrow \arg\max_{v \in \mathcal{V}^0 \setminus \mathcal{L}} r_v$
12
13 **while** $r_{\tilde{v}} > 1$ **do**
14 $\quad \mathcal{L} \leftarrow \mathcal{L} \cup \{\tilde{v}\}$
15 $\quad$ **for** $p_{uw} \in \mathcal{S}_{\tilde{v}}$ **do**
16 $\quad\quad d(p_{uw}) \leftarrow \tilde{v}$
17
18 $\quad$ **for** $v \in V^0 \setminus \mathcal{L}$ **do**
19 $\quad\quad \mathcal{S}_v \leftarrow \mathcal{S}_v \setminus \mathcal{S}_{\tilde{v}}$
20 $\quad\quad \mathcal{V}_v^+ \leftarrow \{u \in \mathcal{V}^+ \mid \exists w \in \mathcal{V}^- \text{ such that } p_{uw} \in \mathcal{S}_v\}$
21 $\quad\quad \mathcal{V}_v^- \leftarrow \{w \in \mathcal{V}^- \mid \exists u \in \mathcal{V}^+ \text{ such that } p_{uw} \in \mathcal{S}_v\}$
22 $\quad\quad r_v \leftarrow \frac{|\mathcal{S}_v|}{\max\{1, |\mathcal{V}_v^+| + |\mathcal{V}_v^-|\}}$
23 $\quad \tilde{v} \leftarrow \arg\max_{v \in V^0 \setminus \mathcal{L}} r_v$
24
25 **return** $\mathcal{I}^{\mathcal{S},\mathcal{L},d}$

4.2.7. Variable Bounds for L^1 Instances

Considering the KKT reformulation for MaxTP in Subsection 4.2.5, we are going to derive corresponding variable bounds for the L^1 instances in this section. We start with the arc flow variables f.

Lemma 5. *Let $\mathcal{I}^{\mathcal{S},\mathcal{L},d}$ be an L^1 instance of some MaxTP instance $\mathcal{I}$. There exists an optimal solution for $\mathcal{I}^{\mathcal{S},\mathcal{L},d}$ such that $f_a \leq \overline{f}_a$, where $\overline{f}_a$ is defined as*

$$\begin{aligned}
\overline{f}_a &:= \min\{\overline{b}_u, |\underline{b}_w|\} && \text{if } a = (u,w) \in \mathcal{A} \text{ with } u \in \mathcal{V}^+, w \in \mathcal{V}^-,\\
\overline{f}_a &:= \min\{\overline{b}_u, \textstyle\sum_{w:d(p_{uw})=v} |\underline{b}_w|\} && \text{if } a = (u,v) \in \mathcal{A} \text{ with } u \in \mathcal{V}^+, v \in \mathcal{L},\\
\overline{f}_a &:= \min\{\textstyle\sum_{u:d(p_{uw})=v} \overline{b}_u, |\underline{b}_w|\} && \text{if } a = (v,w) \in \mathcal{A} \text{ with } v \in \mathcal{L}, w \in \mathcal{V}^-.
\end{aligned}$$

Proof. This follows from Corollary 4. □

Next, we derive bounds for the π- and ϕ-variables. Therefore, we introduce some additional notation. Consider an L^1 instance $\mathcal{I}^{\mathcal{S},\mathcal{L},d}$. For each $v \in \mathcal{L}$ we denote the set of entries, for which there exists a shortest path $p_{uw} \in \mathcal{S}$ such that $v \in p_{uw}$ by $\mathcal{V}_v^+ := \{u \in \mathcal{V}^+ \,|\, \exists w \in \mathcal{V}^- \text{ such that } d(p_{uw}) = v\}$. The analogous set of exits we denote by $\mathcal{V}_v^- := \{w \in \mathcal{V}^- \,|\, \exists u \in \mathcal{V}^+ \text{ such that } d(p_{uw}) = v\}$. Further, given a feasible solution (b, f), by $\mathcal{V}_{\text{act}}^+ := \{u \in \mathcal{V}^+ \,|\, b_u > 0\} \subseteq \mathcal{V}^+$, $\mathcal{V}_{\text{act}}^- := \{w \in \mathcal{V}^+ \,|\, b_w < 0\} \subseteq \mathcal{V}^-$, and $\mathcal{L}_{\text{act}} := \{v \in \mathcal{L} \,|\, \sum_{u \in \mathcal{V}_v^+} f_{uv} > 0\} \subseteq \mathcal{L}$ we denote the sets of active entries, exits, and layer nodes, respectively, i.e., nodes that are incident to at least one arc with nonzero flow. We can now define the following parameters

$$\begin{aligned}
\overline{\pi}_w &:= \max\{d_{uw} \,|\, u \in \mathcal{V}^+\} \text{ for each } w \in \mathcal{V}^-,\\
\underline{\pi}_w &:= \min\{d_{uw} \,|\, u \in \mathcal{V}^+\} \text{ for each } w \in \mathcal{V}^-,\\
\overline{\pi}_u &:= \max\{\overline{\pi}_w - d_{uw} \,|\, w \in \mathcal{V}^-\} \text{ for each } u \in \mathcal{V}^+,\\
\overline{\pi}_v &:= \max\{\overline{\pi}_w - d_{vw} \,|\, w \in \mathcal{V}_v^-\} \text{ for each } v \in \mathcal{L},\\
\underline{\pi}_v &:= \min\{d_{uv} \,|\, u \in \mathcal{V}_v^+\} \text{ for each } v \in \mathcal{L},\\
\overline{\overline{\pi}}_v &:= \min\{\overline{\pi}_u + d_{uv} \,|\, u \in \mathcal{V}_v^+\} \text{ for each } v \in \mathcal{L},\\
\underline{\underline{\pi}}_v &:= \max\{\underline{\pi}_w - d_{vw} \,|\, w \in \mathcal{V}_v^-\} \text{ for each } v \in \mathcal{L},\\
\overline{\phi}_{uw} &:= d_{uw} - \underline{\pi}_w + \overline{\pi}_u \text{ for each } (u,w) \in \mathcal{A} \text{ with } u \in \mathcal{V}^+, w \in \mathcal{V}^-,\\
\overline{\phi}_{uv} &:= d_{uv} - \underline{\pi}_v + \overline{\pi}_u \text{ for each } (u,v) \in \mathcal{A} \text{ with } u \in \mathcal{V}^+, v \in \mathcal{L},\\
\overline{\phi}_{vw} &:= d_{vw} - \underline{\pi}_w + \overline{\pi}_v \text{ for each } (v,w) \in \mathcal{A} \text{ with } v \in \mathcal{L}, w \in \mathcal{V}^-,\\
\overline{\overline{\phi}}_{uv} &:= d_{uv} - \underline{\underline{\pi}}_v + \overline{\pi}_u \text{ for each } (u,v) \in \mathcal{A} \text{ with } u \in \mathcal{V}^+, v \in \mathcal{L},\\
\overline{\overline{\phi}}_{vw} &:= d_{vw} - \underline{\pi}_w + \overline{\overline{\pi}}_v \text{ for each } (v,w) \in \mathcal{A} \text{ with } v \in \mathcal{L}, w \in \mathcal{V}^-.
\end{aligned}$$

Lemma 6. *Let $\mathcal{I}^{\mathcal{S},\mathcal{L},d}$ be an L^1 instance of some MaxTP instance $\mathcal{I}$. Consider the nonlinear programming formulation from Subsection 4.2.5. For each feasible solution (b^*, f^*, π, ϕ) there exists a feasible solution $(b^*, f^*, \pi^*, \phi^*)$ such that*

$$
\begin{aligned}
\pi^*_u &\in [0, \overline{\pi}_u] && \text{for each } u \in \mathcal{V}^+_{act},\\
\pi^*_u &= \overline{\pi}_u && \text{for each } u \in \mathcal{V}^+ \setminus \mathcal{V}^+_{act},\\
\pi^*_w &\in [\underline{\pi}_w, \overline{\pi}_w] && \text{for each } w \in \mathcal{V}^-_{act},\\
\pi^*_w &= \underline{\pi}_w && \text{for each } w \in \mathcal{V}^- \setminus \mathcal{V}^-_{act},\\
\pi^*_v &\in [\underline{\pi}_v, \overline{\pi}_v] && \text{for each } v \in \mathcal{L}_{act},\\
\pi^*_v &\in [\underline{\underline{\pi}}_v, \overline{\overline{\pi}}_v] && \text{for each } v \in \mathcal{L} \setminus \mathcal{L}_{act},\\
\phi^*_{uw} &\in [0, \overline{\phi}_{uw}] && \text{for each } (u,w) \in \mathcal{A} \text{ with } u \in \mathcal{V}^+ \text{ and } w \in \mathcal{V}^-,\\
\phi^*_{uv} &\in [0, \overline{\phi}_{uv}] && \text{for each } (u,v) \in \mathcal{A} \text{ with } u \in \mathcal{V}^+ \text{ and } v \in \mathcal{L}_{act},\\
\phi^*_{uv} &\in [0, \overline{\overline{\phi}}_{uv}] && \text{for each } (u,v) \in \mathcal{A} \text{ with } u \in \mathcal{V}^+ \text{ and } v \in \mathcal{L} \setminus \mathcal{L}_{act},\\
\phi^*_{vw} &\in [0, \overline{\phi}_{vw}] && \text{for each } (v,w) \in \mathcal{A} \text{ with } v \in \mathcal{L}_{act} \text{ and } w \in \mathcal{V}^-,\\
\phi^*_{vw} &\in [0, \overline{\overline{\phi}}_{vw}] && \text{for each } (v,w) \in \mathcal{A} \text{ with } v \in \mathcal{L} \setminus \mathcal{L}_{act} \text{ and } w \in \mathcal{V}^-
\end{aligned}
$$

with the same objective value. In particular, there exists an optimal solution respecting the bounds stated above.

Proof. Let (b^*, f^*, π, ϕ) be a feasible solution and let $\pi_{\min} := \min\{\pi_u \mid u \in \mathcal{V}^+_{\text{act}}\}$. Consider $(b^*, f^*, \pi^*, \phi^*)$ defined as

$$
\pi^*_v := \begin{cases} \pi_v - \pi_{\min} & v \in \mathcal{V}^+_{\text{act}} \cup \mathcal{V}^-_{\text{act}} \cup \mathcal{L} \\ \underline{\pi}_w & v = w \in \mathcal{V}^- \setminus \mathcal{V}^-_{\text{act}} \\ \overline{\pi}_u & v = u \in \mathcal{V}^+ \setminus \mathcal{V}^+_{\text{act}} \end{cases} \quad \text{and} \quad \phi^*_{xy} := \begin{cases} 0 & f^*_{v_1v_2} > 0 \\ d_{v_1v_2} + \pi^*_{v_1} - \pi^*_{v_2} & f^*_{v_1v_2} = 0. \end{cases}
$$

Since the arc flow variable values are not changed, the objective value remains the same. Furthermore, we have $\pi^*_u \geq 0$ for all $u \in \mathcal{V}^+$ and $\pi^*_{u_0} = 0$ for some $u_0 \in \mathcal{V}^+_{\text{act}}$ by construction. Using this, we can show that the variable bounds stated in the Lemma are respected.

First, consider $w \in \mathcal{V}^-_{\text{act}}$. If $d(p_{u_0w}) = 0$ and therefore $(u_0, w) \in \mathcal{A}$, we have

$$
\begin{aligned}
\pi^*_w &= \pi^*_w - \pi^*_{u_0} \\
&= \pi_w - \pi_{\min} - (\pi_{u_0} - \pi_{\min}) \\
&= \pi_w - \pi_{u_0} \\
&\leq \pi_w - \pi_{u_0} + \phi_{u_0w} \\
&= \ell_{u_0w} = d_{u_0w} \leq \overline{\pi}_w.
\end{aligned}
$$

Otherwise, there exists some $v \in \mathcal{L}$ such that $d(p_{u_0 w}) = v$ and we have

$$\begin{aligned}
\pi_w^* &= \pi_w^* - \pi_v + \pi_v - \pi_{u_0}^* \\
&= (\pi_w - \pi_{\min}) - \pi_v + \pi_v - (\pi_{u_0} - \pi_{\min}) \\
&= \pi_w - \pi_v + \pi_v - \pi_{u_0} \\
&\leq \pi_w - \pi_v + \phi_{vw} + \pi_v - \pi_{u_0} + \phi_{u_0 v} \\
&= \ell_{vw} + \ell_{u_0 v} = d_{u_0 w} \leq \overline{\pi}_w.
\end{aligned}$$

On the other hand, since $w \in \mathcal{V}_{\text{act}}^-$ there exists some $u \in \mathcal{V}_{\text{act}}^+$ such that flow is sent from u towards w. If $d(p_{uw}) = 0$ and therefore $(u, w) \in \mathcal{A}$, we have $f_{uw}^* > 0$ and $\phi_{uw} = 0$, which implies

$$\begin{aligned}
\pi_w^* &\geq \pi_w^* - \pi_u^* \\
&= \pi_w^* - \pi_u^* + \phi_{uw} \\
&= (\pi_w - \pi_{\min}) - (\pi_u - \pi_{\min}) + \phi_{uw} \\
&= \pi_w - \pi_u + \phi_{uw} \\
&= \ell_{uw} = d_{uw} \geq \underline{\pi}_w.
\end{aligned}$$

Otherwise, there exists some $v \in \mathcal{L}$ with $d(p_{uw}) = v$, $f_{uv}^* > 0$ and $f_{vw}^* > 0$, as well as $\phi_{uv} = 0$ and $\phi_{vw} = 0$, which implies

$$\begin{aligned}
\pi_w^* &\geq \pi_w^* - \pi_u^* \\
&= \pi_w^* - \pi_v + \pi_v - \pi_u^* \\
&= (\pi_w - \pi_{\min}) - \pi_v + \pi_v - (\pi_u - \pi_{\min}) \\
&= \pi_w - \pi_v + \phi_{vw} + \pi_v - \pi_u + \phi_{uv} \\
&= \ell_{vw} + \ell_{uv} = d_{uw} \geq \underline{\pi}_w.
\end{aligned}$$

Thus, we have $\pi_w^* \in [\underline{\pi}_w, \overline{\pi}_w]$ for all $w \in \mathcal{V}_{\text{act}}^-$. Furthermore, since $\pi_w^* = \underline{\pi}_w$ for all $w \in \mathcal{V}^- \setminus \mathcal{V}_{\text{act}}^-$, it holds that $\pi_w^* \in [\underline{\pi}_w, \overline{\pi}_w]$ for all $w \in \mathcal{V}^-$.

Second, for each $u \in \mathcal{V}_{\text{act}}^+$ there exists some $w \in \mathcal{V}_{\text{act}}^-$ such that flow is sent from u towards w. If $d(p_{uw}) = 0$, then $f_{uw}^* > 0$ and $\phi_{uw} = 0$ implying that

$$\begin{aligned}
\pi_u^* &= \pi_w^* - \pi_w^* + \pi_u^* \\
&= \pi_w^* - (\pi_w - \pi_{\min}) + (\pi_u - \pi_{\min}) \\
&= \pi_w^* - \pi_w + \pi_u \\
&= \pi_w^* - \pi_w + \pi_u - \phi_{uw} \\
&= \pi_w^* - \ell_{uw} \\
&= \pi_w^* - d_{uw} \\
&\leq \overline{\pi}_w - d_{uw} = \overline{\pi}_u.
\end{aligned}$$

Otherwise, there exists $v \in \mathcal{L}$ with $d(p_{uw}) = v$, $f_{uv}^* > 0$ and $f_{vw}^* > 0$, as well as

$\phi_{uv} = 0$ and $\phi_{vw} = 0$, yielding

$$\begin{aligned}
\pi_u^* &= \pi_w^* - \pi_w^* + \pi_v - \pi_v + \pi_u^* \\
&= \pi_w^* - (\pi_w - \pi_{\min}) + \pi_v - \pi_v + (\pi_u - \pi_{\min}) \\
&= \pi_w^* - \pi_w + \pi_v - \pi_v + \pi_u \\
&= \pi_w^* - \pi_w + \pi_v - \phi_{vw} - \pi_v + \pi_u - \phi_{uv} \\
&= \pi_w^* - \ell_{vw} - \ell_{uv} \\
&= \pi_w^* - d_{uw} \\
&\leq \overline{\pi}_w - d_{uw} \leq \overline{\pi}_u.
\end{aligned}$$

Thus, we have $\pi_u^* \in [0, \overline{\pi}_u]$ for all $u \in \mathcal{V}_{\text{act}}^+$. Furthermore, since we set $\pi_u^* = \overline{\pi}_u$ for all $u \in \mathcal{V}^+ \setminus \mathcal{V}_{\text{act}}^+$, it holds that $\pi_u^* \in [0, \overline{\pi}_u]$ for all $u \in \mathcal{V}^+$.

Third, for $v \in \mathcal{L}$ and $u \in \mathcal{V}_v^+$ we have

$$\begin{aligned}
\pi_v^* &= \ell_{uv} + \pi_u^* - \phi_{uv} \\
&\leq \ell_{uv} + \pi_u^* \\
&\leq \ell_{uv} + \overline{\pi}_u \\
&= d_{uv} + \overline{\pi}_u \leq \overline{\overline{\pi}}_v.
\end{aligned}$$

On the other hand, for $v \in \mathcal{L}$ and $w \in \mathcal{V}_v^-$ we have

$$\begin{aligned}
\pi_v^* &= \pi_w^* - \ell_{vw} + \phi_{vw} \\
&\geq \pi_w^* - \ell_{vw} \\
&\geq \underline{\pi}_w - \ell_{vw} \\
&= \underline{\pi}_w - d_{vw} \geq \underline{\underline{\pi}}_v.
\end{aligned}$$

Thus, we have $\pi_v^* \geq 0$ and $\pi_v^* \in [\underline{\underline{\pi}}_v, \overline{\overline{\pi}}_v]$ for all $v \in \mathcal{L}$.

Moreover, if an inner node is active, i.e. for any $v \in \mathcal{L}_{\text{act}}$, we can tighten these bounds. In this particular case, there exists some $u \in \mathcal{V}_{\text{act}}^+$ such that $u \in \mathcal{V}_v^+$, $f_{uv}^* > 0$ and $\phi_{uv} = 0$, which implies

$$\begin{aligned}
\pi_v^* &\geq \pi_v^* - \pi_u^* \\
&= (\pi_v - \pi_{\min}) - (\pi_u + \pi_{\min}) \\
&= \pi_v - \pi_u \\
&= \pi_v - \pi_u - \phi_{uv} \\
&= \ell_{uv} \\
&= d_{uv} \geq \underline{\pi}_v.
\end{aligned}$$

On the other hand, by flow conservation there exists $w \in \mathcal{V}_{\text{act}}^-$ such that $w \in \mathcal{V}_v^-$,

$f^*_{vw} > 0$, and $\phi_{vw} = 0$, implying

$$\begin{aligned}\pi^*_v &= \pi^*_w - \pi^*_w + \pi^*_v \\ &= \pi^*_w - (\pi_w - \pi_{\min}) + (\pi_v - \pi_{\min}) \\ &= \pi^*_w - \pi_w + \pi_v \\ &= \pi^*_w - \pi_w + \pi_v - \phi_{vw} \\ &= \pi^*_w - \ell_{vw} \\ &= \pi^*_w - d_{vw} \\ &\leq \overline{\pi}_w - d_{vw} \leq \overline{\pi}_v.\end{aligned}$$

Thus, $\pi^*_v \in [\underline{\pi}_v, \overline{\pi}_v]$ for all $v \in \mathcal{L}_{\text{act}}$. Finally, we show that these bounds are actually tighter than the previous ones, i.e., $\overline{\overline{\pi}}_v \geq \overline{\pi}_v \geq \underline{\pi}_v \geq \underline{\underline{\pi}}_v$, since

$$\begin{aligned}\overline{\overline{\pi}}_v - \overline{\pi}_v &= \overline{\pi}_u + d_{uv} - (\overline{\pi}_w - d_{vw}) \\ &= \overline{\pi}_u - \overline{\pi}_w + d_{uv} + d_{vw} \\ &\geq \overline{\pi}_u - (\overline{\pi}_w - d_{uw}) \geq 0,\end{aligned}$$

where $u = \arg\min_{u \in \mathcal{V}^+_v} \overline{\pi}_u + d_{uv}$ and $w = \arg\max_{w \in \mathcal{V}^-_v} \overline{\pi}_w - d_{vw}$,

$$\begin{aligned}\underline{\pi}_v - \underline{\underline{\pi}}_v &= d_{uv} - (\underline{\pi}_w - d_{vw}) \\ &= d_{uv} + d_{vw} - \underline{\pi}_w \\ &\geq d_{uw} - \underline{\pi}_w \geq 0,\end{aligned}$$

where $u = \arg\min_{u \in \mathcal{V}^+_v} d_{uv}$ and $w = \arg\max_{w \in \mathcal{V}^-_v} \underline{\pi}_w - d_{vw}$, and

$$\begin{aligned}\overline{\pi}_v - \underline{\pi}_v &= \overline{\pi}_w - d_{vw} - \underline{\pi}_v \\ &\geq d_{uv} + d_{vw} - d_{vw} - \underline{\pi}_v \\ &\geq d_{uv} - \underline{\pi}_v \geq 0\end{aligned}$$

where $w = \arg\max_{w \in \mathcal{V}^-_v} \overline{\pi}_w - d_{vw}$ and we used that $\overline{\pi}_w \geq d_{uw} = d_{uv} + d_{vw}$ for some path $p_{uw} \in \mathcal{S}$ with $d(p_{uw}) = v$.

Next, we focus on the ϕ-variables. First, we show that $\phi^*_{uv} \in [0, d_{uv} - \underline{\pi}_v + \overline{\pi}_u]$ for each $(u, v) \in \mathcal{A}$ with $u \in \mathcal{V}^+$ and $v \in \mathcal{L}_{\text{act}}$. The upper bound holds since

$$\phi^*_{uv} = \ell_{uv} - \pi^*_v + \pi^*_u \leq \ell_{uv} - \underline{\pi}_v + \overline{\pi}_u = d_{uv} - \underline{\pi}_v + \overline{\pi}_u.$$

For the lower bound, we consider the following three cases:

1. If $u \in \mathcal{V}^+_{\text{act}}$ and $v \in \mathcal{L}_{\text{act}}$ with $f^*_{uv} > 0$ we have $\phi^*_{uv} = 0$.

2. If $u \in \mathcal{V}^+_{\text{act}}$ and $v \in \mathcal{L}_{\text{act}}$ with $f^*_{uv} = 0$ we have

$$\begin{aligned}\phi^*_{uv} &= \ell_{uv} - \pi^*_v + \pi^*_u \\ &= \ell_{uv} - (\pi_v - \pi_{\min}) + (\pi_u - \pi_{\min}) \\ &= \ell_{uv} - \pi_v + \pi_u = \phi_{uv} \geq 0.\end{aligned}$$

3. If $u \in \mathcal{V}^+ \setminus \mathcal{V}^+_{\text{act}}$ and $v \in \mathcal{L}_{\text{act}}$ we have $\pi^*_u = \overline{\pi}_u$ implying

$$\begin{aligned}\phi^*_{uv} &= \ell_{uv} - \pi^*_v + \overline{\pi}_u \\ &\geq \ell_{uv} - \overline{\pi}_v + \overline{\pi}_u \\ &= \ell_{uv} - (\overline{\pi}_w - \ell_{vw}) + \overline{\pi}_u \\ &= -\overline{\pi}_w + \ell_{uv} + \ell_{vw} + \overline{\pi}_u \\ &\geq -\overline{\pi}_w + d_{uw} + \overline{\pi}_u \\ &= -(\overline{\pi}_w - d_{uw}) + \overline{\pi}_u \\ &\geq -\overline{\pi}_u + \overline{\pi}_u = 0,\end{aligned}$$

where $w = \arg\max_{w \in \mathcal{V}^-_v} \overline{\pi}_w - d_{vw}$.

Hence, we have $\phi^*_{uv} \in [0, d_{uv} - \underline{\pi}_v + \overline{\pi}_u]$ for $(u, v) \in \mathcal{A}$ with $u \in \mathcal{V}^+$ and $v \in \mathcal{L}_{\text{act}}$. Next, we show that $\phi^*_{uv} \in [0, d_{uv} - \underline{\underline{\pi}}_v + \overline{\pi}_u]$ for $(u, v) \in \mathcal{A}$ with $u \in \mathcal{V}^+$ and $v \in \mathcal{L} \setminus \mathcal{L}_{\text{act}}$. The upper bound holds since

$$\phi^*_{uv} = \ell_{uv} - \pi^*_v + \pi^*_u \leq \ell_{uv} - \underline{\underline{\pi}}_v + \overline{\pi}_u = d_{uv} - \underline{\underline{\pi}}_v + \overline{\pi}_u.$$

For the lower bound, we consider two cases:

1. If $u \in \mathcal{V}^+ \setminus \mathcal{V}^+_{\text{act}}$ and $v \in \mathcal{L} \setminus \mathcal{L}_{\text{act}}$ we have $\pi^*_u = \overline{\pi}_u$

$$\begin{aligned}\phi^*_{uv} &= \ell_{uv} - \pi^*_v + \overline{\pi}_u \\ &\geq \ell_{uv} - \overline{\overline{\pi}}_v + \overline{\pi}_u \\ &= (\overline{\pi}_u + d_{uv}) - \overline{\overline{\pi}}_v \geq 0.\end{aligned}$$

2. If $u \in \mathcal{V}^+_{\text{act}}$ and $v \in \mathcal{L} \setminus \mathcal{L}_{\text{act}}$ we have

$$\begin{aligned}\phi^*_{uv} &= \ell_{uv} - \pi^*_v + \pi^*_u \\ &= \ell_{uv} - (\pi_v - \pi_{\min}) + (\pi_u - \pi_{\min}) \\ &= \ell_{uv} - \pi_v + \pi_{\min} + \pi_u - \pi_{\min} \\ &= \ell_{uv} - \pi_v + \pi_u = \phi_{uv} \geq 0.\end{aligned}$$

Hence, $\phi^*_{uv} \in [0, d_{uv} - \underline{\underline{\pi}}_v + \overline{\pi}_u]$ for $u \in \mathcal{V}^+$ and each $v \in \mathcal{L} \setminus \mathcal{L}_{\text{act}}$.

Next we show, that the $\phi^*_{vw} \in [0, d_{vw} - \underline{\pi}_w + \overline{\pi}_v]$ for each $(v, w) \in \mathcal{A}$ with $w \in \mathcal{V}^-$

and $v \in \mathcal{L}_{\text{act}}$. The upper bound holds since

$$\phi^*_{vw} = \ell_{vw} - \pi^*_w + \pi^*_v \leq \ell_{vw} - \underline{\pi}_w + \overline{\pi}_v = d_{vw} - \underline{\pi}_w + \overline{\pi}_v.$$

For the lower bound, we consider the following three cases:

1. If $v \in \mathcal{L}_{\text{act}}$ and $w \in \mathcal{V}^-_{\text{act}}$ with $f^*_{vw} > 0$ we have $\phi^*_{vw} = 0$.
2. If $v \in \mathcal{L}_{\text{act}}$ and $w \in \mathcal{V}^-_{\text{act}}$ with $f^*_{vw} = 0$ we have

$$\begin{aligned}\phi^*_{vw} &= \ell_{vw} - \pi^*_w + \pi^*_v \\ &= \ell_{vw} - (\pi_w - \pi_{\min}) + (\pi_v - \pi_{\min}) \\ &= \ell_{vw} - \pi_w + \pi_{\min} + \pi_v - \pi_{\min} \\ &= \ell_{vw} - \pi_w + \pi_v = \phi_{vw} \geq 0.\end{aligned}$$

3. If $v \in \mathcal{L}_{\text{act}}$ and $w \in \mathcal{V}^- \setminus \mathcal{V}^-_{\text{act}}$ we have $\pi^*_w = \underline{\pi}_w$

$$\begin{aligned}\phi^*_{vw} &= \ell_{vw} - \underline{\pi}_w + \pi^*_v \\ &\geq \ell_{vw} - \underline{\pi}_w + \underline{\pi}_v \\ &= (d_{uv} + d_{vw}) - \underline{\pi}_w \\ &\geq d_{uw} - \underline{\pi}_w \geq 0,\end{aligned}$$

   where $u = \arg\max_{u \in \mathcal{V}^+_v} d_{uv}$.

Hence, $\phi^*_{vw} \in [0, d_{vw} - \underline{\pi}_w + \overline{\pi}_v]$ for each $(v, w) \in \mathcal{A}$ with $w \in \mathcal{V}^-$ and $v \in \mathcal{L}_{\text{act}}$. Next we show, that the $\phi^*_{vw} \in [0, d_{vw} - \underline{\pi}_w + \overline{\overline{\pi}}_v]$ for each $(v, w) \in \mathcal{A}$ with $w \in \mathcal{V}^-$ and $v \in \mathcal{L} \setminus \mathcal{L}_{\text{act}}$. The upper bound holds since

$$\phi^*_{vw} = \ell_{vw} - \pi^*_w + \pi^*_v \leq \ell_{vw} - \underline{\pi}_w + \overline{\overline{\pi}}_v = d_{vw} - \underline{\pi}_w + \overline{\overline{\pi}}_v.$$

For the lower bound, we consider the following two cases:

1. If $v \in \mathcal{L} \setminus \mathcal{L}_{\text{act}}$ and $w \in \mathcal{V}^- \setminus \mathcal{V}^-_{\text{act}}$ we have $\pi^*_w = \underline{\pi}_w$ implying

$$\begin{aligned}\phi^*_{vw} &= \ell_{vw} - \underline{\pi}_w + \pi^*_v \\ &\geq \ell_{vw} - \underline{\pi}_w + \underline{\overline{\pi}}_v \\ &= \underline{\overline{\pi}}_v - (\underline{\pi}_w - d_{vw}) \geq 0.\end{aligned}$$

2. If $v \in \mathcal{L} \setminus \mathcal{L}_{\text{act}}$ and $w \in \mathcal{V}^-_{\text{act}}$ we have

$$\begin{aligned}\phi^*_{vw} &= \ell_{vw} - \pi^*_w + \pi^*_v \\ &= \ell_{vw} - (\pi_w - \pi_{\min}) + (\pi_v - \pi_{\min}) \\ &= \ell_{vw} - \pi_w + \pi_{\min} + \pi_v - \pi_{\min} \\ &= \ell_{vw} - \pi_w + \pi_v = \phi_{vw} \geq 0.\end{aligned}$$

Hence, $\phi^*_{vw} \in [0, d_{vw} - \underline{\pi}_w + \overline{\pi}_v]$ for each $(v,w) \in \mathcal{A}$ with $w \in \mathcal{V}^-$ and $v \in \mathcal{L} \setminus \mathcal{L}_{\text{act}}$.

Next we show, that the $\phi^*_{uw} \in [0, \ell_{uw} - \underline{\pi}_w + \overline{\pi}_u]$ for each $(u,w) \in \mathcal{A}$ with $u \in \mathcal{V}^+$ and $w \in \mathcal{V}^-$. We consider five cases:

1. If $u \in \mathcal{V}^+_{\text{act}}$ and $w \in \mathcal{V}^-_{\text{act}}$ with $f^*_{uw} > 0$ we have $\phi^*_{uw} = 0$.

2. If $u \in \mathcal{V}^+_{\text{act}}$ and $w \in \mathcal{V}^-_{\text{act}}$ with $f^*_{uw} = 0$ we have

$$\begin{aligned}
\phi^*_{uw} &= \ell_{uw} - \pi^*_w + \pi^*_u \leq d_{uw} - \underline{\pi}_w + \overline{\pi}_u, \\
\phi^*_{uw} &= \ell_{uw} - \pi^*_w + \pi^*_u \\
&= \ell_{uw} - (\pi_w - \pi_{\min}) + (\pi_u - \pi_{\min}) \\
&= \ell_{uw} - \pi_w + \pi_{\min} + \pi_u - \pi_{\min} \\
&= \ell_{uw} - \pi_w + \pi_u = \phi_{uw} \geq 0.
\end{aligned}$$

3. If $u \in \mathcal{V}^+_{\text{act}}$ and $w \in \mathcal{V}^- \setminus \mathcal{V}^-_{\text{act}}$ we have $\pi^*_w = \underline{\pi}_w$ implying

$$\begin{aligned}
\phi^*_{uw} &= \ell_{uw} - \underline{\pi}_w + \pi^*_u \leq d_{uw} - \underline{\pi}_w + \overline{\pi}_u, \\
\phi^*_{uw} &= \ell_{uw} - \underline{\pi}_w + \pi^*_u \geq \ell_{uw} - \underline{\pi}_w \geq 0.
\end{aligned}$$

4. If $u \in \mathcal{V}^+ \setminus \mathcal{V}^+_{\text{act}}$ and $w \in \mathcal{V}^-_{\text{act}}$ we have $\pi^*_u = \overline{\pi}_u$ implying

$$\begin{aligned}
\phi^*_{uw} &= \ell_{uw} - \pi^*_w + \overline{\pi}_u \leq d_{uw} - \underline{\pi}_w + \overline{\pi}_u \\
\phi^*_{uw} &= \ell_{uw} - \pi^*_w + \overline{\pi}_u \\
&\geq \ell_{uw} - \overline{\pi}_w + \overline{\pi}_u \\
&= \overline{\pi}_u - (\overline{\pi}_w - \ell_{uw}) \geq 0.
\end{aligned}$$

5. If $u \in \mathcal{V}^+ \setminus \mathcal{V}^+_{\text{act}}$ and $w \in \mathcal{V}^- \setminus \mathcal{V}^-_{\text{act}}$ we have $\pi^*_u = \overline{\pi}_u$ and $\pi^*_w = \underline{\pi}_w$ implying

$$\begin{aligned}
\phi^*_{uw} &= \ell_{uw} - \underline{\pi}_w + \overline{\pi}_u = d_{uw} - \underline{\pi}_w + \overline{\pi}_u \\
\phi^*_{uw} &= \ell_{uw} - \underline{\pi}_w + \overline{\pi}_u \\
&\geq \ell_{uw} - \overline{\pi}_w + \overline{\pi}_u \\
&= \overline{\pi}_u - (\overline{\pi}_w - \ell_{uw}) \geq 0.
\end{aligned}$$

Hence, $\phi^*_{uw} \in [0, d_{uw} - \underline{\pi}_w + \overline{\pi}_u]$ for each $(u,w) \in \mathcal{A}$ with $u \in \mathcal{V}^+$ and $w \in \mathcal{V}^-$.

It remains to show that $(b^*, f^*, \pi^*, \phi^*)$ is feasible. The dual constraints (4.12) as well as the complementarity conditions (4.13) are satisfied by construction. Additionally, the bounds proven above show that the nonnegativity conditions (4.16) are respected, too. Finally, since we left the f^*- and b^*-variables unchanged, this shows that $(b^*, f^*, \pi^*, \phi^*)$ is a feasible solution. □

4.2.8. A MIP Model for L^1 Instances and Valid Inequalities

Using the bounds from the previous section, i.e., Lemma 5 and Lemma 6, and applying the big-M technique to the KKT reformulation from Subsection 4.2.5, we derive the following MIP model for an L^1 instance $\mathcal{I}^{\mathcal{S},\mathcal{L},d} = (\mathcal{V}, \mathcal{A}, \ell, \underline{b}, \bar{b})$ derived from a MaxTP instance $\mathcal{I}$.

$$\max_{b,f,\phi,\pi,x} \quad \sum_{a\in\mathcal{A}} \ell_a f_a \tag{4.23}$$

$$\text{s.t.} \quad \sum_{u\in\mathcal{V}^+} b_u + \sum_{w\in\mathcal{V}^-} b_w = 0 \tag{4.24}$$

$$\sum_{a\in\delta^+(v)} f_a - \sum_{a\in\delta^-(v)} f_a = b_v \qquad \forall v \in \mathcal{V}^+ \cup \mathcal{V}^- \tag{4.25}$$

$$\sum_{a\in\delta^+(v)} f_a - \sum_{a\in\delta^-(v)} f_a = 0 \qquad \forall v \in \mathcal{V}^0 \tag{4.26}$$

$$\pi_v - \pi_u + \phi_a = \ell_a \qquad \forall (u,v) = a \in \mathcal{A} \tag{4.27}$$

$$f_a \le \overline{f}_a x_a \qquad \forall a \in \mathcal{A} \tag{4.28}$$

$$\phi_a \le \overline{\overline{\phi}}_a (1 - x_a) \qquad \forall a \in \mathcal{A} \tag{4.29}$$

$$b_v \in [\underline{b}_v, \bar{b}_v] \qquad \forall v \in \mathcal{V}^+ \cup \mathcal{V}^- \tag{4.30}$$

$$f_a \in [0, \overline{f}_a] \qquad \forall a \in \mathcal{A} \tag{4.31}$$

$$\phi_a \in [0, \overline{\overline{\phi}}_a] \qquad \forall a \in \mathcal{A} \tag{4.32}$$

$$\pi_u \in [0, \overline{\pi}_u] \qquad \forall u \in \mathcal{V}^+ \tag{4.33}$$

$$\pi_w \in [\underline{\pi}_w, \overline{\pi}_w] \qquad \forall w \in \mathcal{V}^- \tag{4.34}$$

$$\pi_v \in [\underline{\underline{\pi}}_v, \overline{\overline{\pi}}_v] \qquad \forall v \in \mathcal{L} \tag{4.35}$$

$$x_a \in \{0,1\} \qquad \forall a \in \mathcal{A} \tag{4.36}$$

Besides adapting the variable bounds, we introduce an auxiliary binary variable $x_a \in \{0,1\}$ for each arc $a \in \mathcal{A}$ here to apply the big-M technique to the complementarity conditions (4.13). This enables us to replace (4.13) by the two linear constraints (4.28) and (4.29). Depending on the value of x_a, either $f_a = 0$ or $\phi_a = 0$, which ensures that (4.13) is satisfied.

Activity-Related Bound-Tightening

To enforce the tighter bounds for active layer nodes, we introduce additional auxiliary variables $y_v \in \{0,1\}$ for each $v \in \mathcal{L}$ indicating whether v is active or not through constraints

$$y_v \ge x_a \quad \forall v \in \mathcal{L}, \forall a \in \delta^-(v) \cup \delta^+(v). \tag{4.37}$$

If there is nonzero flow on some arc $a \in \delta^-(v) \cup \delta^+(v)$, we have $x_a = 1$ due to constraints (4.28) and $y_v = 1$ due to constraints (4.37). Thus, using y_v, we can tighten the π_v-variable and the ϕ_a-variables via constraints

$$\pi_v \geq \underline{\underline{\pi}}_v - (\underline{\underline{\pi}}_v - \underline{\pi}_v)y_v \qquad \forall v \in \mathcal{L} \tag{4.38}$$

$$\pi_v \leq \overline{\overline{\pi}}_v - (\overline{\overline{\pi}}_v - \overline{\pi}_v)y_v \qquad \forall v \in \mathcal{L} \tag{4.39}$$

$$\phi_a \leq \overline{\overline{\phi}}_a - (\overline{\overline{\phi}}_a - \overline{\phi}_a)y_v \qquad \forall v \in \mathcal{L}, \forall a \in \delta^-(v) \cup \delta^+(v). \tag{4.40}$$

Forcing the π_v-variable of an inactive entry or exit $v \in (\mathcal{V}^+ \setminus \mathcal{V}^+_{\text{act}}) \cup (\mathcal{V}^- \setminus \mathcal{V}^-_{\text{act}})$ to be equal to its upper or lower bound, respectively, is more difficult. In particular, we cannot rely on the x_a-variables here, because, if we consider some source $u \in \mathcal{V}^+$, then $x_a = 1$ for some $a \in \delta^+(u)$ does not necessarily imply $f_a > 0$ and the activity of u as the only adjacent source.

However, if all supply and demand intervals have integer bounds, i.e., if $\bar{b}_v \in \mathbb{Z}$ and $\underline{b}_v \in \mathbb{Z}$ for all $v \in \mathcal{V}^+ \cup \mathcal{V}^-$, then there exists an optimal solution such that $b_v \in \mathbb{Z}$ for all $v \in \mathcal{V}^+ \cup \mathcal{V}^-$ due to the bound-closeness result in Lemma 1. In particular, it holds that $b_u \geq 1$ and $b_w \leq -1$ for active entries and exits, respectively. Thus, constraints

$$y_u \leq b_u \qquad \forall u \in \mathcal{V}^+ \tag{4.41}$$

$$-y_w \geq b_w \qquad \forall w \in \mathcal{V}^- \tag{4.42}$$

enforce the desired behavior in this case. To enforce the tighter bounds for π_v-variables that correspond to non-active entries or exits $v \in (\mathcal{V}^+ \setminus \mathcal{V}^+_{\text{act}}) \cup (\mathcal{V}^- \setminus \mathcal{V}^-_{\text{act}})$, we then add the following inequalities:

$$\pi_u \geq \overline{\pi}_u - \overline{\pi}_u y_u \qquad \forall u \in \mathcal{V}^+ \tag{4.43}$$

$$\pi_w \leq \underline{\pi}_w - (\underline{\pi}_w - \overline{\pi}_w)y_w \qquad \forall w \in \mathcal{V}^-. \tag{4.44}$$

No-Detour Constraints

Finally, another set of valid inequalities is motivated by the following observation: Let $\mathcal{I}^{\mathcal{S},\mathcal{L},d}$ be an L^1 derived from some MaxTP instance $\mathcal{I}$. Further, let $v \in \mathcal{L}$, $u \in \mathcal{V}^+_v$ and $w \in \mathcal{V}^-_v$, i.e., we have $(u, v) \in \mathcal{A}$ and $(v, w) \in \mathcal{A}$. If $d(p_{uw}) \neq v$, i.e., the shortest path $p_{uw} \in \mathcal{S}$ has not been assigned to v, there exists another path between u and w in the corresponding L^1 network, which is not longer than $\ell_{uv} + \ell_{vw}$. Hence, the following no-detour constraints, which ensure that $f_{uv} = 0$ or $f_{vw} = 0$, can be added without decreasing the objective value of an optimal solution

$$x_{uv} + x_{vw} \leq 1 \quad \forall v \in \mathcal{L}, \forall u \in \mathcal{V}^+_v, \text{ and } \forall w \in \mathcal{V}^-_v \text{ with } d(p_{uw}) \neq v. \tag{4.45}$$

4.2.9. Side Note: The Minimum Transportation Problem (MinTP)

A natural question to ask when investigating worst case scenarios is what best case scenarios look like. Thus, in this section, we discuss the Minimum Transportation Problem (MinTP), whose goal is, in contrast to MaxTP, to find a supply and demand vector such that the optimal objective value of the induced TP instance is minimized.

Theorem 2. MinTP *can be solved in polynomial time.*

Given a MinTP instance $\mathcal{I} = (\mathcal{V}, \mathcal{A}, \ell, \bar{b}, \underline{b})$, we define a corresponding MCF instance $\mathcal{I}^{\text{MCF}} = (\mathcal{V}^{\text{MCF}}, \mathcal{A}^{\text{MCF}}, \ell^{\text{MCF}}, c, b)$ as follows. Let

$$B^+_{\min} := \sum_{v \in \mathcal{V}^+} \underline{b}_v, \quad B^+_{\max} := \sum_{v \in \mathcal{V}^+} \bar{b}_v, \quad B^-_{\min} := \sum_{v \in \mathcal{V}^-} |\bar{b}_v|, \quad B^-_{\max} := \sum_{v \in \mathcal{V}^-} |\underline{b}_v|$$
$$B_{\min} := \max\{B^+_{\min}, B^-_{\min}\}, \quad B_{\max} := \min\{B^+_{\max}, B^-_{\max}\}.$$

$B^+_{\min}$ denotes the minimum and $B^+_{\max}$ denotes the maximum possible amount of supply w.r.t. the corresponding bounds. Analogously, $B^-_{\min}$ and $B^-_{\max}$ are the minimum and maximum absolute demand. Consequently, $B_{\min}$ and $B_{\max}$ represent the minimum and maximum possible amount of flow to enter and leave the network.

The vertex set $\mathcal{V}^{\text{MCF}}$ is equal to $\mathcal{V}$ together with four additional vertices, i.e., we set $\mathcal{V}^{\text{MCF}} := \mathcal{V} \cup \{s, s', t', t\}$. Thereby, s serves as the only source, while t represents the only sink of the network. We set $b_s := B_{\max}$ and $b_t := -B_{\max}$ as supply and demand value, respectively.

Next, we describe the composition of the arc set $\mathcal{A}^{\text{MCF}}$. First, we add a copy of each $a \in \mathcal{A}$ together with the corresponding length value. We still consider these arcs to be uncapacitated and therefore define $c_a := \min\{\sum_{u \in \mathcal{V}^+} \bar{b}_u, \sum_{w \in \mathcal{V}^-} |\underline{b}_w|\}$. These copied arcs we denote by $\mathcal{A}^1$ in the following.

Additionally, we add an arc $a = (s, u)$ from s towards each vertex corresponding to a source $u \in \mathcal{V}^+$ in $\mathcal{I}$ with capacity $c_a := \underline{b}_u$. This set of arcs we denote by $\mathcal{A}^2 := \{(s, u) \mid u \in \mathcal{V}^+\}$. Analogously, we add an arc $a = (s', u)$ from s' towards each vertex corresponding to a source $u \in \mathcal{V}^+$ in $\mathcal{I}$ with capacity $c_a := \bar{b}_u - \underline{b}_u$. In the following, we denote these arcs by $\mathcal{A}^3 := \{(s', u) \mid u \in \mathcal{V}^+\}$. Similarly, we add an arc $a = (w, t)$ from each vertex corresponding to a sink $w \in \mathcal{V}^-$ in $\mathcal{I}$ towards t with capacity $c_a := |\bar{b}_w|$. This set of arcs we denote by $\mathcal{A}^4 := \{(w, t) \mid w \in \mathcal{V}^-\}$. Additionally, we add an arc $a = (w, t')$ from each vertex corresponding to a sink $w \in \mathcal{V}^-$ in $\mathcal{I}$ towards t' with capacity $c_a := |\underline{b}_w| - |\bar{b}_w|$. This arc set we denote by $\mathcal{A}^5 := \{(w, t') \mid w \in \mathcal{V}^-\}$.

Finally, we add three more arcs to the network. First, an arc $a_1 = (s, s')$ with capacity $c^{\text{MCF}}_{a_1} := B_{\max} - B^+_{\min}$. Second, an arc $a_2 = (t', t)$ with capacity $c^{\text{MCF}}_{a_2} := B_{\max} - B^-_{\min}$. Third, an arc $a_3 = (s', t')$ with capacity $c^{\text{MCF}}_{a_3} := B_{\max} - B_{\min}$. Finally, for all $a \in \mathcal{A}^{\text{MCF}} \setminus \mathcal{A}^1$ we set $\ell^{\text{MCF}}_a := 0$. The MCF instance $\mathcal{I}^{\text{MCF}}$ corresponding to the example MinTP instance $\mathcal{I}$ from Figure 4.11 together with an optimal solution are shown in Figure 4.12.

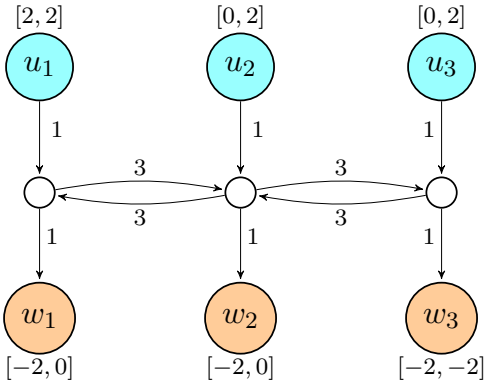
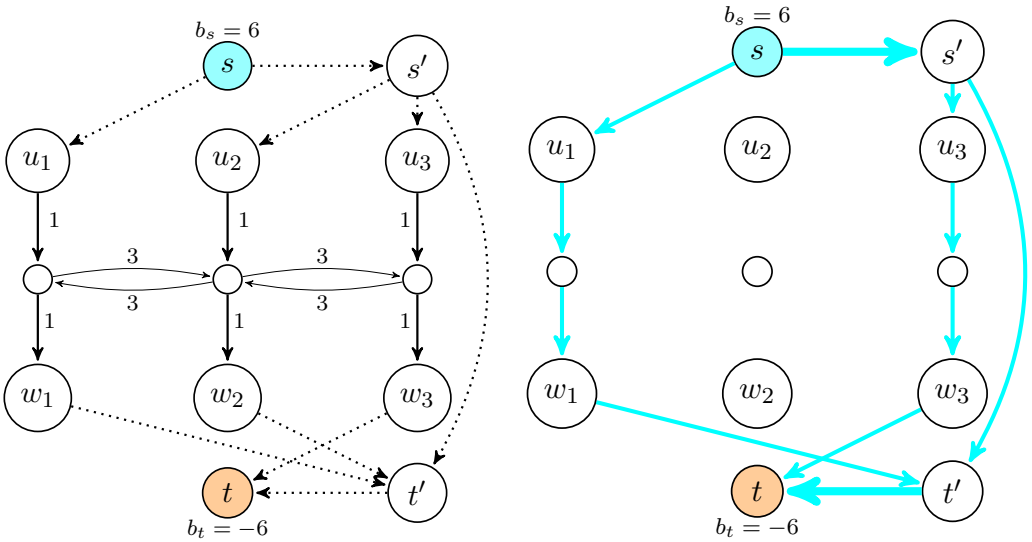


Figure 4.11.: Example MinTP instance with the supply and demand intervals above and below the corresponding nodes and the lengths ℓ_a written next to the corresponding arcs.



(a) MCF instance $\mathcal{I}^{\text{MCF}}$ for the MinTP instance $\mathcal{I}$ from Figure 4.11.

(b) An optimal solution for $\mathcal{I}^{\text{MCF}}$ with objective value 8.

Figure 4.12.: Figure 4.12a shows MCF instance $\mathcal{I}^{\text{MCF}}$ corresponding to the example MinTP instance $\mathcal{I}$ from Figure 4.11. All arcs with zero capacity, which would have to be added, are not shown here. On the other hand, all newly added arcs with nonzero capacity are dotted. They have a capacity value of $c_a = 2$, except for $(s, s'), (s', t)$, and (t', t), for which we have $c_a = 4$, and zero length, i.e., $\ell_a = 0$. The solid arcs remain uncapacitated. Figure 4.12b shows an optimal solution for $\mathcal{I}^{\text{MCF}}$. The two thick blue arcs (s, s') and (t', t) represent a flow value of $f_a = 4$, while all other blue arcs have a flow value of $f_a = 2$. The invisible arcs carry no flow. The cost of the flow is $\sum_{a \in \mathcal{A}^{\text{MCF}}} \ell_a^{\text{MCF}} f_a = 8$.

Lemma 7. *There exists a one-to-one correspondence between the solutions of the MinTP instance $\mathcal{I}$ and the feasible solutions of the corresponding MCF instance $\mathcal{I}^{MCF}$ preserving the objective value.*

Proof. Let (b, f) be a solution for $\mathcal{I}$. Then

$$\tilde{f}_a := \begin{cases} f_a & \text{if } a \in A_1 \\ \underline{b}_u & \text{if } a = (s,u) \in A_2 \\ b_u - \underline{b}_u & \text{if } a = (s',u) \in A_3 \\ |\bar{b}_w| & \text{if } a = (w,t) \in A_4 \\ |b_w| - |\bar{b}_w| & \text{if } a = (w,t') \in A_5 \\ B_{\max} - B^+_{\min} & \text{if } a = a_1 \\ B_{\max} - B^-_{\min} & \text{if } a = a_2 \\ B_{\max} - \sum\limits_{u \in \mathcal{V}} b_u & \text{if } a = a_3 \end{cases}$$

is a feasible solution for MCF instance $\mathcal{I}^{\text{MCF}}$. The flow values for the arcs in $\mathcal{A}^2$ and $\mathcal{A}^4$ as well as for a_1 and a_2 do not depend on (b, f). In fact, these values have to be the same in all feasible solutions of $\mathcal{I}^{\text{MCF}}$ to satisfy the supply and demand of s and t, respectively. Further, the flow values of A_1 directly correspond to the flow values f, and the flow values for the arcs in $\mathcal{A}^3$ and $\mathcal{A}^5$ depend on the supply and demand of the corresponding sources $u \in \mathcal{V}^+$ and sinks $w \in \mathcal{V}^-$, respectively. Similarly, the flow on arc a_3 depends on the total supply. By construction, all flow values respect the capacities and flow conservation is ensured. Thus, there are no two pairwise different feasible solutions for $\mathcal{I}$ that are mapped on the same feasible solution for $\mathcal{I}^{\text{MCF}}$.

On the other hand, let $\tilde{f}_a$ be a feasible solution for MCF instance $\mathcal{I}^{\text{MCF}}$. Then

$$b_v := \begin{cases} \tilde{f}_{su} + \tilde{f}_{s'u} & \text{for } v = u \in \mathcal{V}^+ \\ -\tilde{f}_{wt} - \tilde{f}_{wt'} & \text{for } v = w \in \mathcal{V}^- \end{cases}$$

and

$$f_a := \tilde{f}_a \qquad \text{for } a \in \mathcal{A}$$

denotes a solution for MinTP instance $\mathcal{I}$. By construction, flow conservation is ensured at the inner nodes, the supplies and demands of all sources and sinks are satisfied, respectively, and the corresponding interval bounds are respected. Additionally, as argued above, only the flow values of the arcs in $\mathcal{A}^1, \mathcal{A}^2$ and $\mathcal{A}^4$ are not fixed and any change in one of the corresponding variable values leads to a different solution (b, f). Thus, there are no two pairwise different feasible solutions for $\mathcal{I}^{\text{MCF}}$ that map towards a common solution for $\mathcal{I}$.

Finally, only the arcs in $\mathcal{A}^1$ in $\mathcal{I}^{\text{MCF}}$ can have nonzero length by construction. Thus, since the flow values on these arcs are preserved by the bijection induced by the two mappings above, the objective value is preserved. □

Corollary 7. *There exists a one-to-one mapping between the optimal solutions of MinTP instance $\mathcal{I}$ and the optimal solutions of MCF instance $\mathcal{I}^{MCF}$.*

Theorem 2. *MinTP can be solved in polynomial time.*

Proof. Creating MCF instance $\mathcal{I}^{\mathrm{MCF}}$, which has $|\mathcal{V}^{\mathrm{MCF}}| = |\mathcal{V}| + 4$ vertices and $|\mathcal{A}^{\mathrm{MCF}}| = |\bigcup_{i=1}^{5} A_i \cup \{a_1, a_2, a_3\}| = |\mathcal{A}| + 2|\mathcal{V}^+| + 2|\mathcal{V}^-| + 3$ arcs, solving it using any polynomial-time algorithm for MCF, and applying the mapping defined in the proof of Lemma 7, yields an optimal solution for $\mathcal{I}$ in polynomial time. □

Another way to derive this result is to adapt the bilevel formulation for MaxTP from Section 4.2.4 to model MinTP. To do this, we change the objective sense of the upper-level objective function (4.1) from max to min. Since the objective functions of both levels are identical, the formulation can be transformed into a single-level linear program by lifting the lower-level constraints.

4.3. The Maximum Potential Transport Moment Problem (MaxPTM)

In this section, we propose an alternative way to distribute the flow on the arcs as a basis for introducing another severity measure w.r.t. gas transport scenarios. While the flow on the arcs is distributed to lead to a minimum transport moment in MaxTP, i.e., the objective value of the corresponding TP instances, this point of view may be overly simplistic. In particular, in MaxTP only shortest paths in the network are used, while alternative routes are not considered.

As discussed in Section 3.5.2, gas flow through pipelines causes a friction induced pressure loss. In the stationary case [96, 81], this is often modeled using the well-known Weymouth equation, and for a pipeline $a = (u, v)$, it can be stated as

$$p_u^2 - p_v^2 = \ell_a C_a q_a |q_a|.$$

Here, p_u and p_v denote the pressures at the corresponding nodes, q_a is the mass flow in the pipe, and the coefficient C_a incorporates associated pipeline parameters such as the diameter and the integral roughness. These equations imply that mass flow cannot go in directed cycles and is distributed among several paths from the entries to the exits depending on the "resistance" given by the coefficients ℓ_a and C_a. Furthermore, in passive networks, which consist of pipelines only and contain no active elements, the flow distribution q is uniquely determined by the supplies and demands, see [26, 51] for details.

Instead of using the nonlinear Weymouth equation, we employ a similar, but linear model. In accordance with the definition of the minimum transport moment, we only consider the pipeline length as resistance coefficient here. For a pipeline $a = (u, v)$, our model reads

$$\pi_u - \pi_v = \ell_a (f_a^{\mathrm{fwd}} - f_a^{\mathrm{bwd}}).$$

In this equation, two nonnegative flow variables are used, which represent flow going in the forward and backward direction, respectively. Thus, we assume that flow in both directions is possible on each arc in the following, which is a natural assumption for passive networks. Nevertheless, the equations still satisfy the conditions that lead to a unique flow distribution f, which means that no objective function must be considered when routing the flow through the network. This constitutes a significant difference compared to MaxTP.

Next, we formulate the optimization problem to find a most severe transport situation w.r.t. what we call the potential transport moment. In particular, the Maximum Potential Transport Moment Problem (MaxPTM) can be stated as the following nonlinear program

$$\max_{f,\pi,b} \sum_{a\in\mathcal{A}} \ell_a f_a \tag{4.46}$$

$$\sum_{a\in\delta^+(v)} (f_a^{\text{fwd}} - f_a^{\text{bwd}}) - \sum_{a\in\delta^-(v)} (f_a^{\text{fwd}} - f_a^{\text{bwd}}) = b_v \qquad \forall v \in \mathcal{V}^+ \cup \mathcal{V}^- \tag{4.47}$$

$$\sum_{a\in\delta^+(v)} (f_a^{\text{fwd}} - f_a^{\text{bwd}}) - \sum_{a\in\delta^-(v)} (f_a^{\text{fwd}} - f_a^{\text{bwd}}) = 0 \qquad \forall v \in \mathcal{V}^0 \tag{4.48}$$

$$\pi_u - \pi_v - \ell_a(f_a^{\text{fwd}} - f_a^{\text{bwd}}) = 0 \qquad \forall a \in \mathcal{A} \tag{4.49}$$

$$f_a^{\text{fwd}} f_a^{\text{bwd}} = 0 \qquad \forall a \in \mathcal{A} \tag{4.50}$$

$$f_a^{\text{fwd}}, f_a^{\text{bwd}} \in \mathbb{R}_{\geq 0} \qquad \forall a \in \mathcal{A} \tag{4.51}$$

$$b_v \in [\underline{b}_v, \bar{b}_v] \qquad \forall v \in \mathcal{V}^+ \cup \mathcal{V}^- \tag{4.52}$$

$$\pi_v \in \mathbb{R} \qquad \forall v \in \mathcal{V}. \tag{4.53}$$

The complementarity constraints (4.50) are necessary because flow would otherwise be sent in cycles on the arcs, making the problem unbounded. However, (4.50) can be reformulated using the big-M technique. Therefore, for each $a \in \mathcal{A}$, we introduce an auxiliary binary variable $x_a \in \{0, 1\}$ and replace (4.50) with

$$f_a^{\text{fwd}} \leq \overline{f} x_a \tag{4.54}$$

$$f_a^{\text{bwd}} \leq \overline{f}(1 - x_a). \tag{4.55}$$

Here, $\overline{f} := \min\{\sum_{u\in\mathcal{V}^+} \bar{b}_u, \sum_{w\in\mathcal{V}^-} |\underline{b}_w|\}$ denotes the maximum possible flow into and out of the network. Further, in contrast to MaxTP, the x_a-variable indicate whether flow goes in forward or backward direction on arc $a \in \mathcal{A}$.

4.3.1. Complexity of MaxPTM

Due to the similar concepts behind MaxTP and MaxPTM, the following result is not a surprise.

Theorem 3. MaxPTM *is NP-hard.*

We use an adaption of the proof for MaxTP here, compare Theorem 1 in Section 4.2.3. In particular, we again reduce from Partition, and given an instance of it, we create the identical flow network with the same parameters.

Since the arcs in $\mathcal{A}^1$ have zero length, it follows that u_i and w_i have the same node potential, i.e., we have $\pi_{u_i} = \pi_{w_i}$ for all $i \in \{1, \dots, n\}$ due to constraints (4.49). Furthermore, it is easy to see that Lemma 2 and Corollary 1 hold for MaxPTM, too, i.e., for every feasible (optimal) solution there exists a feasible (optimal) solution with complementary supply and demand vector and the same objective value. Additionally, we assume w.l.o.g. that $\pi_v = 0$ holds in every feasible solution, since an arbitrary constant could be added to all node potentials. Thus, given a balanced and complementary supply and demand value b, the corresponding unique induced feasible solution is given by

$$\pi_v := \begin{cases} \frac{b_{u_i} + b_{w_i}}{2} & \text{if } v = u_i \in \mathcal{V}^+ \text{ or } w_i \in \mathcal{V}^- \\ 0 & \text{otherwise ,} \end{cases}$$

$$f_a^{\rightarrow} := \begin{cases} \frac{b_{u_i} + |b_{w_i}|}{2} & \text{if } a = (u_i, w_i) \in \mathcal{A}^1 \\ \frac{b_{u_i}}{2} & \text{if } a = (u_i, v) \in \mathcal{A}^2 \\ \frac{|b_{w_i}|}{2} & \text{if } a = (v, w_i) \in \mathcal{A}^3 \end{cases} \quad \text{and} \quad f_a^{\leftarrow} := \begin{cases} 0 & \text{if } a = (u_i, w_i) \in \mathcal{A}^1 \\ \frac{|b_{w_i}|}{2} & \text{if } a = (u_i, v) \in \mathcal{A}^2 \\ \frac{b_{u_i}}{2} & \text{if } a = (v, w_i) \in \mathcal{A}^3. \end{cases}$$

Further, we can show that Lemma 3 holds for MaxPTM, too. Therefore, we add a different intermediate step when summing up the flows on the arcs and putting them into relation to the sum of supplies and absolute demands, i.e., we use

$$\begin{aligned} c(b, f) &= \sum_{(u_i, v) \in \mathcal{A}^2} (f^{\rightarrow}_{u_i v} + f^{\leftarrow}_{u_i v}) + \sum_{(v, w_i) \in \mathcal{A}^3} (f^{\rightarrow}_{v w_i} + f^{\leftarrow}_{v w_i}) \\ &= \Big(\sum_{(u_i, v) \in \mathcal{A}^2} f^{\rightarrow}_{u_i v} + \sum_{(v, w_i) \in \mathcal{A}^3} f^{\leftarrow}_{v w_i} \Big) + \Big(\sum_{(u_i, v) \in \mathcal{A}^2} f^{\leftarrow}_{u_i v} + \sum_{(v, w_i) \in \mathcal{A}^3} f^{\rightarrow}_{v w_i} \Big) \\ &= \sum_{z_i \in Z'} b_{u_i} + \sum_{z_i \in Z \setminus Z'} |b_{w_i}| \end{aligned}$$

in the arguments of both directions of the proof. Hence, MaxPTM is NP-hard, too.

4.4. Case Study: gaslib-582

To conclude this chapter, we present a computational study based on the data from the `gaslib-582` network from the GasLib benchmark library [147]. The network topology and parameters are based on slightly perturbed real-world data of parts of the German gas transport infrastructure and summarized below.

instance	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{V}^+\rvert$	$\lvert\mathcal{V}^-\rvert$
gaslib-582	582	1170	15	70

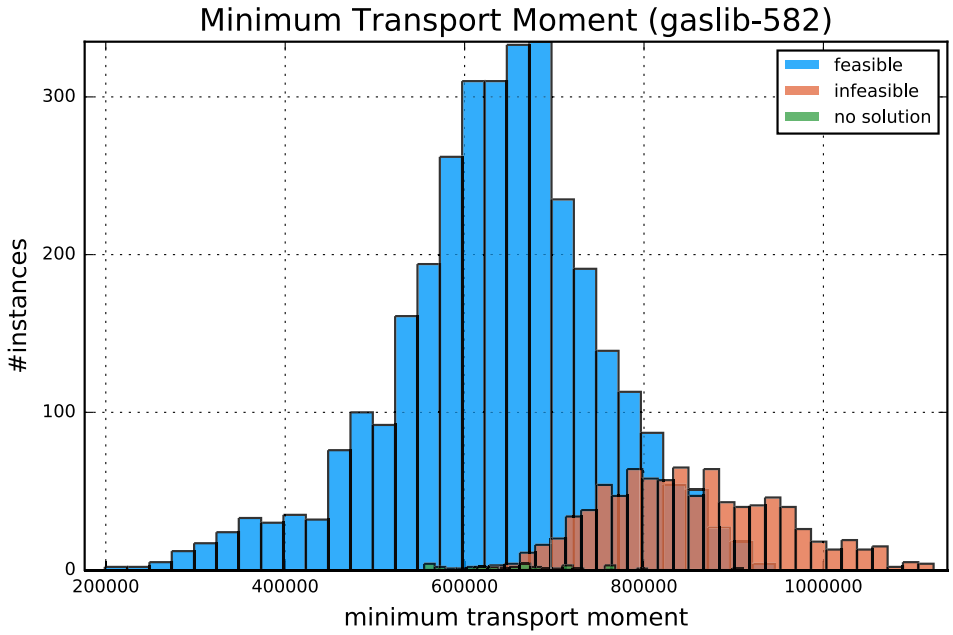


Figure 4.13.: Histogram of minimum transport moment for transport scenarios in gaslib-582, grouped by feasibility status.

In addition, it contains a collection of 4227 balanced transport scenarios, which were created with the methods described in Chapter 14 of [96] and should cover realistic scenarios well. For each of them, we know its feasibility status w.r.t. the detailed MIP model in Chapter 6 of [96]. This status can be feasible if a solution for the transport scenario has been found, infeasible if it has been proven that no such solution exists, or no solution, meaning that no feasible solution has been found within the considered time limit, although it might still exist. In this sense, we consider the infeasible supply and demand vectors to be more severe than the feasible ones.

First, we have computed the optimal solution values of the corresponding Transportation Problem instances for all these transport scenarios, i.e., their minimum transport moment. They are shown in the histogram in Figure 4.13, grouped by their feasibility status. The values range from approximately 198847 to 1122826, and the feasible transport scenarios have a smaller optimal objective value than the infeasible ones. However, there is some overlap where transport scenarios with the same minimum transport moment are feasible or infeasible.

Additionally, we computed the potential transport moment for all transport scenarios. Here, the values range from approximately 204260 to 1140021. The corresponding histogram shown in Figure 4.14 looks almost identical.

To decide whether the benchmark library contains severe transport situations w.r.t. our two measures, we solved a MaxTP instance and a MaxPTM instance, which we generated based on the `gaslib-582` data. In particular, for each entry, we

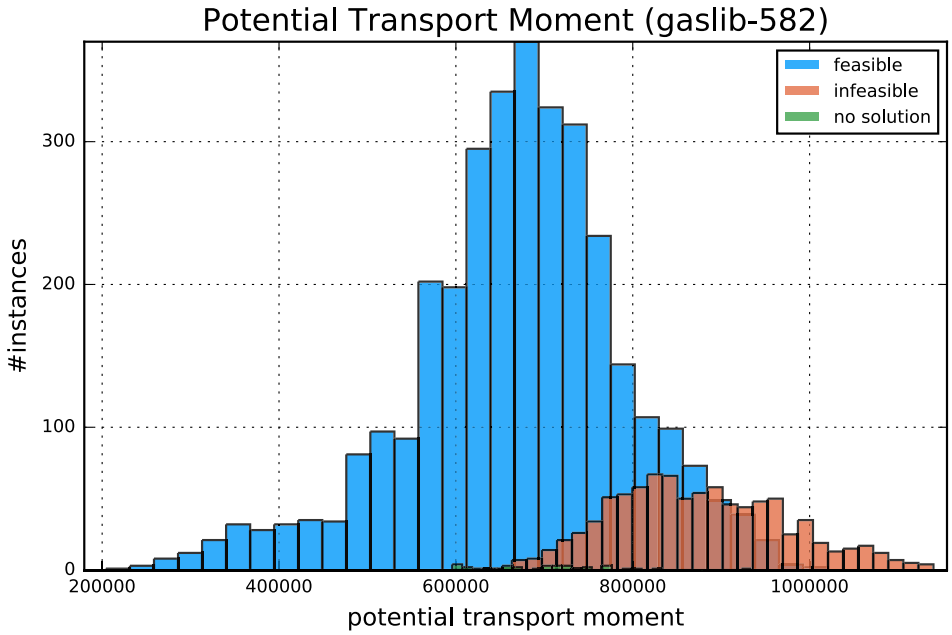


Figure 4.14.: Histogram of potential transport moment for transport scenarios in gaslib-582, grouped by feasibility status.

used zero as the lower and the maximum supply value occurring in these scenarios as upper supply bound. Equivalently, for each exit, we used zero as the upper and the minimum demand value occurring in these scenarios as the lower demand bound. To solve MaxTP, we applied the following method: First, we derived a solution-equivalent L^1 instance using the heuristic in Algorithm 8 described at the end of Section 4.2.6. Afterward, we determined the bounds stated in Subsection 4.2.7 and subsequently solved the corresponding MIP model presented in Section 4.2.8 with all described constraints added. Finally, this solution is mapped back to the original problem using the procedure described in Lemma 4. For MaxPTM, we directly applied the MIP model defined in Section 4.3. Note that both MIPs were solved in less than a second using Gurobi [69].

Solving the MaxTP instance yields an optimal solution having objective value 1406674, which is about 23% larger than the maximum value from the data set. On the other hand, the FDB-MCF heuristic from Subsection 4.2.2 generates a transport scenario with a minimum transport moment of 1379907 and does not find the optimal solution. The optimal solution is visualized in Figure 4.15. Green and orange disks represent entry and exit nodes, respectively, and solid colors show the used capacity while transparent colors show available but unused capacities. The flow on the arcs is drawn in blue. However, note that not many nodes are visible in these drawings. This is because, although the network features 31 entries and 129 exits, only 85 have nonzero supply or demand and often a relatively small amount.

On the other hand, solving the MaxPTM instance yields an optimal solution with

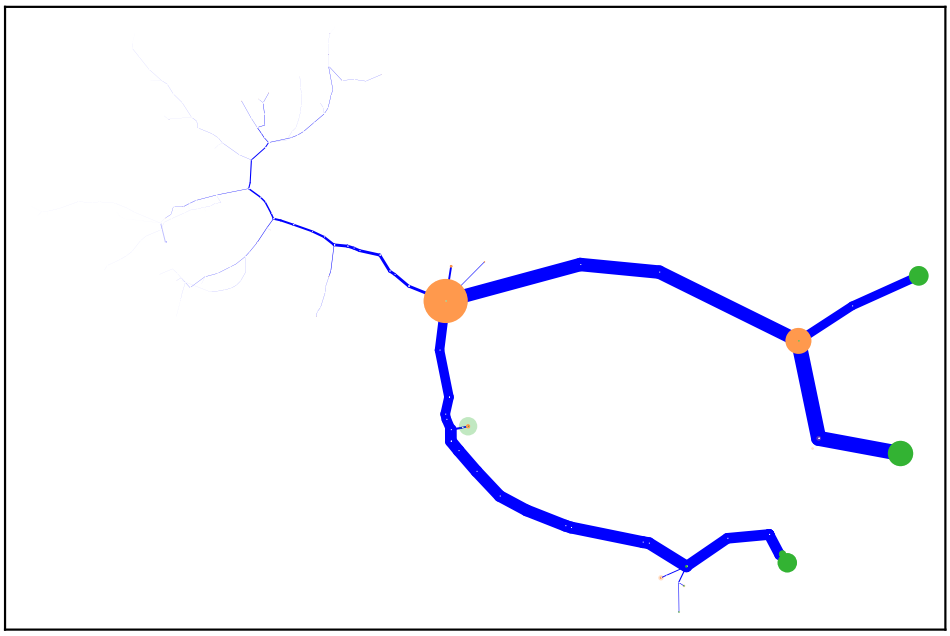


Figure 4.15.: An optimal solution of the MaxTP instance for gaslib-582.

an objective value of 1456905, which is almost 30% larger than the maximum value in the data set. Its optimal solution is visualized in Figure 4.16.

If we calculate the minimum transport moment for the optimal transport scenario for MaxPTM, we derive a value of 1375486. Hence, the two optimal solutions clearly differ in their supply and demand vector. The absolute supply and demand differences as well as the absolute flow differences between the two optimal solutions are visualized in Figure 4.17. Note that flows, demands, and supplies differences are only shown here if they differ by more than 10^{-6}. We observe that many cyclic structures are present. This is because MaxPTM uses multiple paths for the flow between the entries and exits. In contrast, an optimal solution for MaxTP has a forest- or tree-shaped structure w.r.t. the arc flows.

4.5. Conclusion and Future Research

Chapter 4 discusses the critical problem of identifying severe transport scenarios in gas transport. The motivation to consider this problem is that most approaches for determining and evaluating the technical capacity of gas transport networks used in practice are sample-based. Here, transport scenarios that are severe w.r.t. some given measure are identified in a first step. The resulting test sets are then created and designed with the goal in mind to cover and represent all difficult flow situations that can arise. Afterward, the corresponding scenarios are evaluated using highly detailed mathematical models for the physics of gas transport using methods from

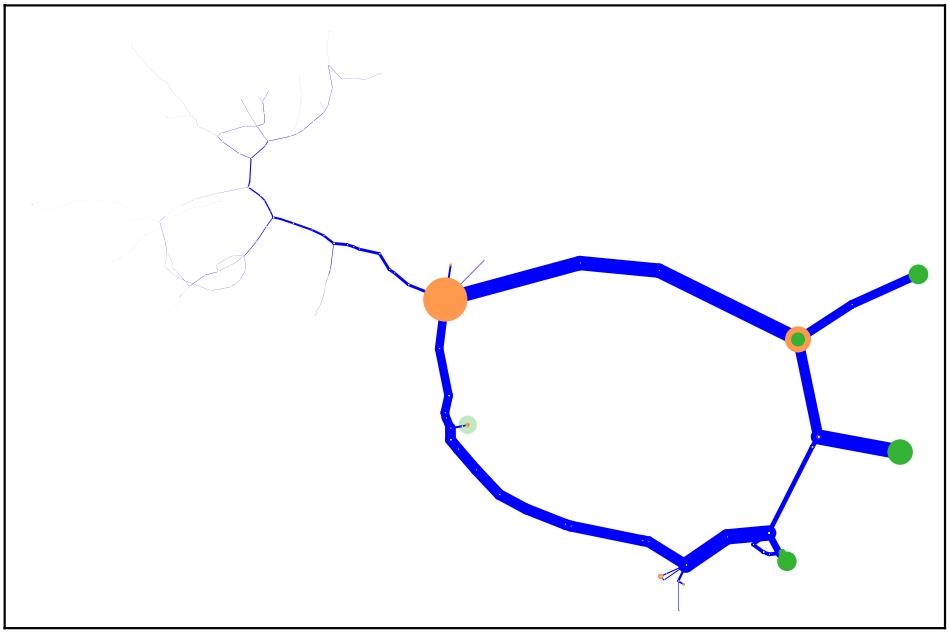


Figure 4.16.: An optimal solution of the MaxPTM instance for gaslib-582.

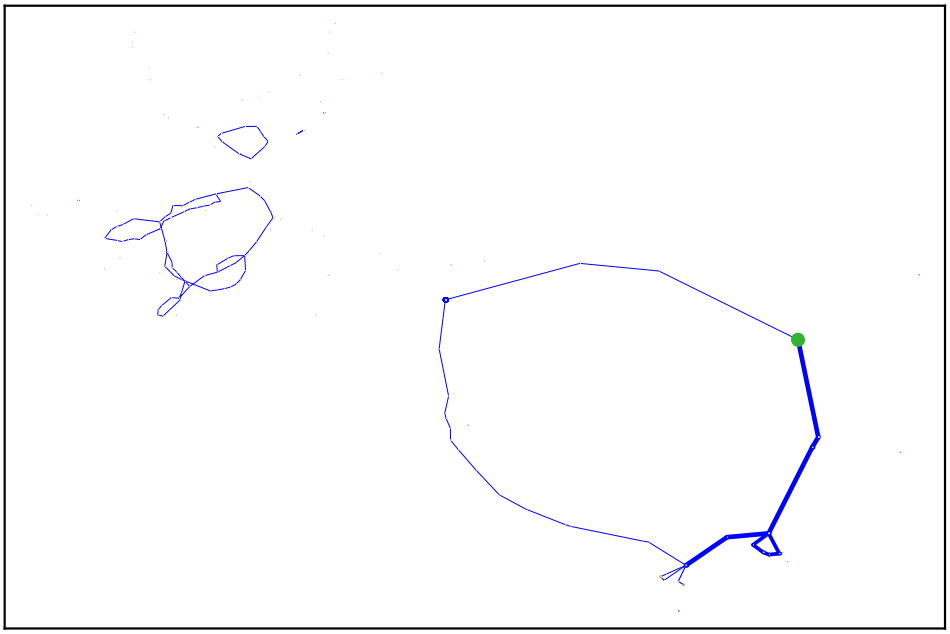


Figure 4.17.: Absolute supply, demand, and flow differences between the two optimal solutions of MaxTP and MaxPTM from Figure 4.15 and Figure 4.16.

optimization or simulation in a second step. If transport can be realized for all or at least a great majority of them, the corresponding technical capacity is considered feasible. Two approaches of this type are discussed in Section 4.1.2.

However, the discussion on the Reference Point Method at the end of Section 4.1.2 and the case study in Section 4.4 demonstrate that currently used test sets do not contain all transport scenarios that should be considered severe. Moreover, it seems reasonable to apply several diverse severity measures to improve coverage. This motivates the introduction of our two new severity measures: The minimum transport moment and the potential transport moment. In this context, the goal of the Maximum Transportation Problem (MaxTP) and the Maximum Potential Transport Moment Problem (MaxPTM), which are introduced in Section 4.2 and Section 4.3, respectively, is to find corresponding worst-case scenarios.

In the real-world entry-exit gas market model, explained in detail in Section 4.1, the shippers nominate the amounts of gas they want to insert into or withdraw from the network the day before transport takes place. Afterward, the TSOs then have to realize transport. MaxTP and MaxPTM are based on the idea of modeling this process as a Stackelberg game. The leader, i.e., the shippers, chooses a balanced transport scenario from a feasible supply and demand polytope, while the follower, i.e., the TSO, subsequently routes the flow through the network according to a given flow model and objective function, which represents the considered severity measure. Thus, to determine worst-case scenarios, the leader chooses a transport scenario leading to a maximum transport effort for the follower, i.e., its goal is to maximize the objective function value of the follower.

According to the game model described above, we introduce a linear bilevel optimization formulation with interdicting objective functions for MaxTP in Subsection 4.2.4. In order to solve it, we apply a classical KKT reformulation in Subsection 4.2.5 and obtain a linear programming model with additional complementarity constraints. To reformulate it as a MIP using the big-M technique, we next introduce the notion of solution-equivalency for MaxTP instances in Subsection 4.2.6. This equivalency relation helps us identify MaxTP instances whose feasible solutions can easily be mapped onto each other while the objective function values are preserved. In particular, we introduce L^1 instances, which are tripartite, and acyclic, and can generically be constructed in a way that they feature a small number of arcs. Their structure allows us to prove bounds for the variables in the corresponding KKT reformulations in Subsection 4.2.7 and to derive a MIP model in Subsection 4.2.8.

The main difference between MaxTP and MaxPTM is the flow model of the follower. Here, MaxTP incorporates the probably most simple model, an uncapacitated linear flow. Thus, we were a little surprised to find out that MaxTP is NP-hard, which is proven in Subsection 4.2.3. For quite some time, we believed that the FDB-MCF heuristic for MaxTP proposed in Subsection 4.2.2 is an exact algorithm and solves the problem since it ensures the bound-closeness of the resulting supply and demand vectors, which is the property of an optimal solution, see Subsection 4.2.1. However, the NP-hardness proof shows that the complexity lies in the decisions made in the upper level, i.e., when choosing an admissible supply and the demand vector.

The flow model plays no role here, which is further underlined by the fact that the corresponding proof can easily be adapted to show the NP-hardness of MaxPTM in Subsection 4.3.1.

In contrast to MaxTP, MaxPTM applies a more elaborate linear potential flow model, which we consider to be more realistic w.r.t. the physics of the gas flow. Moreover, it has the advantage compared to MaxTP that, given a transport scenario, it admits a unique feasible solution w.r.t. the arc flows. Therefore, and since the potential transport moment exclusively relies on these values, MaxPTM can directly be modeled as MIP.

There are several directions for future research. First, considering the bound-closeness result for an optimal solution of MaxTP, the question arises whether this can be exploited to derive additional inequalities for the MIP model. Additionally, it is often beneficial to further tighten the variable bounds from a computational point of view. In this regard, the question arises whether other generic classes of solution-equivalent MaxTP instances exist, which feature an even smaller number of arcs or allow for tighter variable bounds. Such a question seems interesting for another more abstract reason: How large does a network have to be to preserve important information, in our case, the flows between the entry and exit pairs and the objective function value? Although it seems a long stretch, a similar question is of interest regarding the designing neural networks. Thereby, small networks carrying the same information may allow for a higher degree of explainability, which is a current topic of intensive research in the area of machine learning.

Furthermore, due to the straightforward and natural setup of MaxTP as well as its natural generalization, the Maximum Min-Cost-Flow Problem (MaxMCF), see Appendix B.1, there probably exists a vast variety of further real-world applications, which could be modeled using them. For the same reason and because of their simple setup, we also believe that the two problems constitute new and challenging network flow problems on their own, i.e., they are interesting for the scientific community even without a motivating application in mind. The presence of additional capacity constraints in MaxMCF makes the problem probably harder from a computational complexity point of view. However, we have not been able to prove a corresponding result yet. Additionally, it seems more challenging to derive a MIP model for MaxMCF due to the additional classes of dual variables and complementarity constraints, which arise because of the capacity restrictions. Moreover, it is not clear if and how the concept of solution-equivalency can be extended to it. Summing up, we believe that further research regarding these two problems is fruitful w.r.t. potential applications and an exciting mathematical endeavour at the same time.

Finally, returning to the motivating application, using the minimum transport moment as a severity measure still has its limitations. First, the physical details of how the gas flow is distributed among the pipelines and the resulting pressure drop are highly simplified. Here, the potential transport moment can be seen as the first step towards a more detailed view. Another possibility would be to apply the Weymouth equations to model the gas flow in the pipelines. However, this would result in a nonconvex and nonlinear lower-level model. Furthermore, the impact

and the capacity of active network elements such as compressors or valves have been completely ignored so far, even though they play a key role in actual operations. In this regard, a lower bound on the flow throughput does often exist, which is required when using compressors. With this knowledge in mind, we cannot claim in general that if a specific transport scenario is feasible, then all scaled-down scenarios are feasible, too. Finally, another possible severity measure may be the total cost of compression required to realize transport while satisfying all pressure bounds in this context. However, this cost is related to the used compression power, a nonlinear and nonconvex function of pressure ratio and flow.

5. Monitoring Transport Infrastructure

The content of this thesis up until now can briefly be summarized as follows. Chapter 3 focuses on the optimization of the transient control of gas transport networks. We discuss how the gas flow through pipelines and the control of active elements such as compressors, control valves, and valves can be modeled. Based on this, we present a two-stage approach for determining a stable network control that minimizes the usage of technical and non-technical measures.

In Chapter 4, we review the German entry-exit gas market model and explain how transport scenarios arise due to the interaction between the gas traders and the TSOs. Afterward, we introduce new concepts and methods for identifying severe transport scenarios, which is a crucial task in the context of determining a network's technical capacity, i.e., the amounts of gas it can transport.

Let us consider personal computers as a metaphor for gas transport networks for a moment. Up until now, we have mainly talked about the system's software, which is undoubtedly important w.r.t. user experience, working efficiency, and quality of the results. However, as many of us unfortunately experience at some point in our lives, malfunctioning hardware and problems related to it are not only annoying but can also lead to complete system breakdowns. Projecting this image back to gas transport networks, it becomes clear that monitoring its elements is essential in order to prevent serious failures.

Besides the obvious impact on everyday operations, several other aspects w.r.t. a malfunctioning transport infrastructure must be taken into account. Let us study the issue of pipeline leakages. Not only do these result in high monetary costs, but they can even lead to fatalities [88]. Moreover, the energy sector, especially the natural gas and coal industry, is responsible for more than a quarter of anthropogenic methane emissions, and methane as a greenhouse gas has a significant global warming potential [89]. Thus, it is crucial to identify pipeline leaks early on.

Nowadays, monitoring tasks, such as the one implicitly described above, are often performed by uncrewed aerial vehicles (UAVs). This has several advantages. For example, their usage is often cheap because the number of necessary person-hours for operating them is comparatively small. But more importantly, they can gather information about an area from a long distance or high altitude and visit areas that are not accessible in any other way. To see this, consider for example their application in the fight against forest fires, as described by Ollero et al. [122], or the collection of data to analyze widespread animal populations, as suggested by Chamoso et al. [24]. Hence, it is not surprising that Hausamann et al. [73] already discussed the usage of UAVs for monitoring natural gas pipelines in 2005. The authors find that the developments in technology show that they can provide the

appropriate platform for a remote sensing-based inspection system. Moreover, their operation seems technically feasible in controlled and uncontrolled airspace.

The Length-Constrained Cycle Partition Problem (LCCP), which represents the main topic in the remainder of this chapter, constitutes a new routing problem regarding these UAVs and is motivated by applications such as the ones described above. Given a set of areas $\mathcal{V} = \{v_1, \dots, v_n\}$ to monitor, the goal is to determine the minimum number of UAVs necessary to do this. Thereby, the individual flying routes must satisfy the following three conditions: First, each UAV must repeatedly fly the same tour, which means that it starts and ends its route at the same area and visits all other areas assigned to it exactly once. Second, each area is visited by exactly one UAV and therefore contained in exactly one tour. This is required to avoid possible interferences resulting from intersections. Third, each area $v_i \in \mathcal{V}$ is associated with a critical weight value $T_i \in \mathbb{R}_{\geq 0}$, which represents an upper bound on the duration for which it can be left unattended, and a scanning time $S_i \in \mathbb{R}_{\geq 0}$, which is the amount of time a UAV needs to scan it. We require that after scanning v_i for S_i time units, the UAV assigned to it has to return and rescan it within T_i.

A possible way to derive critical weight values in UAV-related applications is the following. Suppose we are given the probability of an undesired event $p_i(t)$ at area v_i, which we assume to grow with the time t that has passed after its last scan. The critical time, i.e., the corresponding weight T_i, is obtained from the equation $p_i(T_i) = \bar{p}_i$ where $\bar{p}_i$ is a threshold for the risk of an undesired event. The value $\bar{p}_i$ could, for example, depend on the expected damage associated with area v_i.

The remainder of this chapter is structured as follows. After giving a proper definition of LCCP in Section 5.1, we discuss previous work on related problems in Section 5.2. Afterward, we introduce preprocessing techniques and the concept of conflict hypergraphs for LCCP in Section 5.3 and Section 5.4, respectively. Furthermore, we present our Most-Critical-Vertex-Heuristic and two MIP models in Section 5.5 and Section 5.6, and conclude with a discussion on our computational experiments as well as an outlook on future research in Section 5.7 and Section 5.8.

5.1. The Length-Constrained Cycle Partition Problem (LCCP)

In this section, we formally introduce the Length-Constrained Cycle Partition Problem (LCCP). As we will discuss later on in detail, LCCP can be seen as a generalization of the well-known Traveling Salesperson Problem (TSP). Therefore, in contrast to the two previous chapters, we apply graph-theoretical nomenclature here.

5.1.1. Problem Definition

For LCCP, we are given an undirected graph $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_1, \dots, v_n\}$ denotes the set of vertices and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the set of edges. For each vertex $v_i \in \mathcal{V}$, we are given a critical weight $T_i \in \mathbb{R}_{\geq 0}$ and a scanning time $S_i \in \mathbb{R}_{\geq 0}$ such that

$S_i \leq T_i$. The weight of an edge $e_{ij} := \{v_i, v_j\} \in \mathcal{E}$ is given by $\hat{L}_{ij} \in \mathbb{R}_{\geq 0}$. Note that in the UAV routing problems described above, $\hat{L}_{ij}$ corresponds to the flying time between the areas v_i and v_j. Finally, an LCCP instance on a complete graph is called metric if the edge weights obey the triangle inequality.

To simplify notation, let us denote a cycle in G as a tuple $C_k = (\mathcal{V}_k, \mathcal{E}_k)$ in the following, where $\mathcal{V}_k$ and $\mathcal{E}_k$ represent its vertices and edges, respectively. Further, we call a cycle C_k proper if $|\mathcal{V}_k| \geq 2$, a singleton if $|\mathcal{V}_k| = 1$, and empty otherwise.

Next, we call a cycle C_k feasible if $\tau_k \leq T_i$ holds for each $v_i \in \mathcal{V}_k$ where

$$\tau_k := \sum_{e_{ij} \in \mathcal{E}_k} \hat{L}_{ij} + \sum_{v_i \in \mathcal{V}_k} S_i$$

denotes the length of C_k, i.e., if the length of a cycle is not greater than the critical weight value of each of its vertices. Furthermore, a solution for LCCP is a cycle partition $\mathcal{C} = \{C_1, \ldots, C_m\}$ of $\mathcal{V}$, i.e., a vertex disjoint cycle cover, and $\mathcal{C}$ is called feasible if all of its cycles are feasible. For example, LCCP always admits the trivial feasible solution consisting of all vertices being singletons. Nevertheless, the goal of LCCP is to determine a feasible cycle partition with a minimum number of cycles.

In the following, w.l.o.g. we assume that $S_i = 0$ for all vertices $v_i \in \mathcal{V}$ since we can add the scanning times to the edge weights. Indeed, let $L_{ij} := \hat{L}_{ij} + \frac{S_i + S_j}{2}$ for $e_{ij} \in \mathcal{E}$ and consider some cycle C_k. Then

$$\tau_k = \sum_{e_{ij} \in \mathcal{E}_k} \hat{L}_{ij} + \sum_{v_i \in \mathcal{V}_k} S_i = \sum_{e_{ij} \in \mathcal{E}_k} \left(\hat{L}_{ij} + \frac{S_i + S_j}{2} \right) = \sum_{e_{ij} \in \mathcal{E}_k} L_{ij}.$$

A metric LCCP instance remains metric under this transformation, i.e., the triangle inequality is preserved. Hence, we finally define an instance of LCCP as a four-tuple $(\mathcal{V}, \mathcal{E}, T, L)$ with $T \in \mathbb{R}_{\geq 0}^{|\mathcal{V}|}$ and $L \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|}$. An example instance illustrating the previous definitions is shown in Figure 5.1.

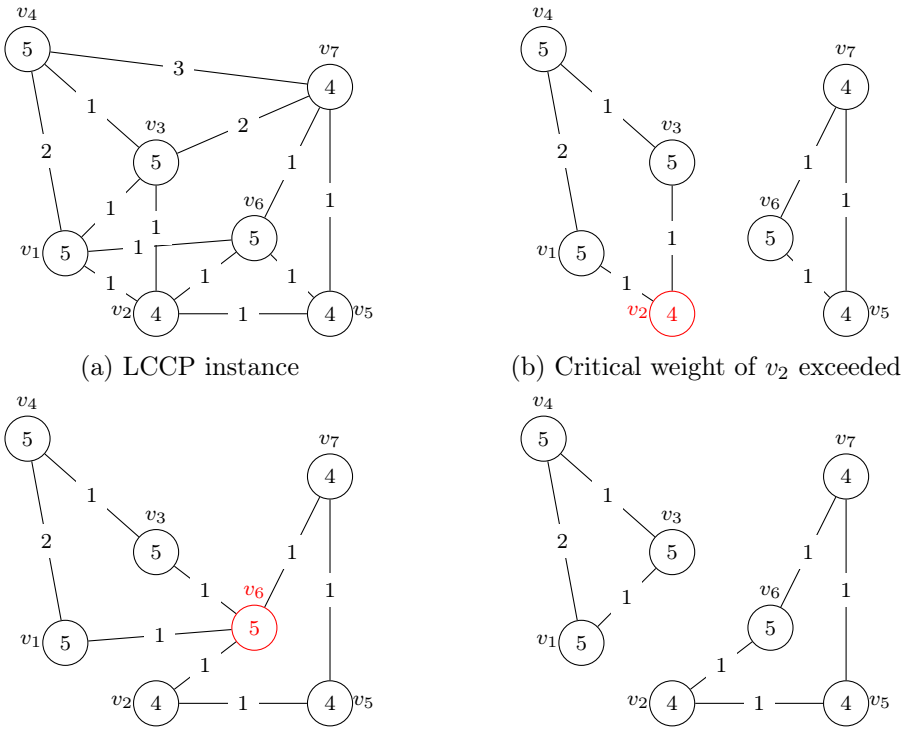
5.1.2. Complexity of LCCP

Besides proving NP-hardness, with the following Lemma, it becomes evident that the Traveling Salesperson Problem (TSP) can be seen as a special case of LCCP.

Lemma 8. *LCCP is NP-hard.*

Proof. Consider an instance $(\mathcal{V}, \mathcal{E}, L)$ of TSP and some $B \geq 0$. Setting $T_i := B$ for each $v_i \in \mathcal{V}$, we create an instance $(\mathcal{V}, \mathcal{E}, T, L)$ of LCCP. An optimal solution for this LCCP instance consists of exactly one tour if and only if there exists a Hamiltonian cycle with length not greater than B. Since the decision variant of TSP is NP-complete, see Garey and Johnson [55], it follows that LCCP is NP-hard. □

This result can further be refined for non-metric instances. Lemma 9 answers the question if a polynomial-time algorithm exists, which is guaranteed to find a solution to LCCP within a constant factor of the size of a minimum cycle partition.



(a) LCCP instance

(b) Critical weight of v_2 exceeded

(c) v_6 contained in two cycles

(d) An optimal solution

Figure 5.1.: LCCP instance with two infeasible and one optimal solution.

Lemma 9. *General LCCP does not belong to the class of APX problems.*

Proof. Consider the Minimum Vertex Disjoint Cycle Cover Problem (MVDCC), which was shown not to be in APX by Sahni and Gonzalez [143]. MVDCC is defined on an unweighted undirected graph $(\mathcal{V}, \mathcal{E})$ and its goal is to determine a minimum vertex disjoint cycle cover. Setting $T_i = n$ for all $v_i \in \mathcal{V}$ and $L_{ij} = 1$ for all $e_{ij} \in \mathcal{E}$ we derive an LCCP instance $(\mathcal{V}, \mathcal{E}, T, L)$. Any constant approximation algorithm for LCCP would induce a constant approximation algorithm for MVDCC. □

However, it should be noted that the question whether metric LCCP is in the class of APX problems or not remains open.

5.2. Related Work

Hoppmann et al. [82] first introduced LCCP, and the results obtained there were extended and generalized in a follow-up paper by the same authors [85]. In this section, we review important work on problems related to it.

Drucker et al. consider the Cyclic Routing of UAVs (CR-UAV) problem in [38]. In CR-UAV, closed walks, which have to start and end at the same vertex but can visit vertices and edges multiple times, must be determined. Additionally, waiting at vertices is possible and different routes are allowed to intersect. In particular, they can share vertices as well as edges. The goal is to determine the minimum number of UAVs necessary to jointly satisfy the critical weight conditions on the vertices. Thus, we note that each feasible solution for the equivalent LCCP instance is a feasible solution for CR-UAV, too. However, the reverse statement does not hold in general. The close relation between CR-UAV and LCCP is further discussed from a computational point of view in Subsection 5.7.5. Ho and Ouaknine [80] show that the decision problem corresponding to CR-UAV is PSPACE-complete, even in the case of a single UAV. In [40], Drucker et al. present a solution approach based on a binary search on the number k of available UAVs and solving satisfiability problems. These satisfiability problems try to answer whether CR-UAV admits a feasible solution for the given k or not. The corresponding formulations use so-called slots, which correspond to arrival times of the UAVs at the vertices. However, since the necessary number of slots is not known in advance and the underlying algorithm answers to a predefined upper bound on them, the proposed method is incomplete, i.e., it is not guaranteed to determine an optimal solution. The authors present computational experiments on instances with up to seven vertices and three UAVs. The corresponding data is provided at [39].

Furthermore, in [37], Drucker et al. suggest a reduction to model-checking and present an exact algorithm, which runs a bounded model checker for detecting feasible solutions and an explicit-state search attempting to prove their absence in parallel. Asghar et al. [5], who synonymously call the problem Multi-Robot Routing for Persistent Monitoring with Latency Constraints, develop a factor $\mathcal{O}(log\,\rho)$ approximation algorithm, where ρ is the ratio of the maximum and the minimum

critical weight. They partition the vertices w.r.t. their critical weights and subsequently solve a Minimum Cycle Cover Problem (MCCP) on each subset, which we discuss in the following paragraph.

Given a graph $G = (\mathcal{V}, \mathcal{E})$ and some λ, the goal of MCCP is to determine the minimum number of cycles covering the vertex set, such that the length of each cycle is not greater than λ. For the metric version of this NP-hard problem, an $\frac{32}{7}$ approximation algorithm is proposed in [165].

Furthermore, as demonstrated and introduced in the proof of Lemma 9, the Minimum Vertex Disjoint Cycle Cover Problem (MVDCC) [143] is a problem from combinatorial optimization, for which LCCP can be seen as a generalization.

Finally, providing a service to a set of customers on a periodical basis is a recurring task inherent to many real-world applications. The corresponding optimization problems, where it is often the goal to determine the minimum number of agents needed to satisfy the service requirements while taking several resource restrictions into account, represent a broad area of active research. Classic examples include the delivery of gasoline to service stations [30] or timetabling in public transport [105, 149].

In particular, besides TSP, which, as we have shown, is a special case of LCCP, several other well-known combinatorial optimization problems of this type are closely related to LCCP. One of them is the Vehicle Routing Problem with Time Windows (VRPTW), see Solomon and Desrosiers [152] or Desrochers et al. [32] for surveys on the topic. The goal is to determine a collection of routes for a fleet of homogeneous vehicles. The routes have to start and end at a common depot v_0 and jointly visit a given set of customers $\{v_1, \dots, v_n\}$. Further, each customer $v_i \in \mathcal{V}$ has some service requirement q_i which has to be satisfied within a time window $[l_i, u_i]$ by precisely one vehicle. One of the most studied objectives for VRPTW is to minimize the number of necessary vehicles while the accumulated requirements of the customers are not allowed to exceed the capacities Q of the assigned vehicles. However, compared to LCCP, a significant difference is that no repetitions of the tours is considered.

5.3. Preprocessing

In this section, we show how edges can be identified, which cannot be part of any feasible cycle. Before proving a corresponding condition, we introduce the concept of a completion for an edge.

Definition 7. *Let $(\mathcal{V}, \mathcal{E}, T, L)$ be an LCCP instance, and let $e_{ij} \in \mathcal{E}$. A completion for e_{ij} is a path p_{ij} between v_i and v_j such that $\ell(p_{ij}) + L_{ij} \leq T_s$ for all $v_s \in p_{ij}$. It is called a shortest completion if it has minimum length among all completions.*

Lemma 10. *Let $(\mathcal{V}, \mathcal{E}, T, L)$ be an LCCP instance, $e_{ij} \in \mathcal{E}$, and p_{ij} be a shortest completion for e_{ij}. Then $\ell(p_{ij}) + L_{ij}$ is a tight lower bound on the length of a proper feasible cycle containing e_{ij}.*

Proof. Let C_k be a feasible cycle of length τ_k such that $e_{ij} \in \mathcal{E}_k$. C_k can be split into e_{ij} and a completion $\hat{p}_{ij}$. Thus,

$$\tau_k = \ell(\hat{p}_{ij}) + L_{ij} \geq \ell(p_{ij}) + L_{ij},$$

and since the cycle induced by p_{ij} and e_{ij} is feasible, the bound is tight. □

Corollary 8. *Let $(\mathcal{V}, \mathcal{E}, T, L)$ be an LCCP instance. If there exists no completion for some edge e_{ij}, there exists no feasible cycle containing it.*

Remark 1. *In the metric case, if an edge $e_{ij} \in \mathcal{E}$ possesses some completion, then e_{ij} is a shortest completion for itself due to the triangle inequality. Hence, in this case, there exists no completion for an edge $e_{ij} \in \mathcal{E}$ if $2L_{ij} > \min\{T_i, T_j\}$.*

The following lemma shows that determining a shortest completion or proving that no completion exists can be done in polynomial time w.r.t. the size of G.

Lemma 11. *Let $(\mathcal{V}, \mathcal{E}, T, L)$ be an LCCP instance, and let $e_{ij} \in \mathcal{E}$. Determining a shortest completion for e_{ij} or proving that no completion exists can be done in $\mathcal{O}(|V||E| + |V|^2 \cdot \log|V|)$.*

Proof. Consider Algorithm 9. In each iteration, a shortest path p_{ij} between v_i and v_j is computed. Next, we check if the critical weight values of all of its vertices are respected. If this is the case, we have found a shortest completion. Otherwise, we remove all vertices from G whose critical weight values are smaller than $\ell(p_{ij}) + L_{ij}$ and continue with the next iteration. The removal of vertices does not decrease the length of a shortest path. Hence, these vertices can never be part of any completion for e_{ij}. If v_i or v_j is removed at some point, no completion exists.

Algorithm 9: Does there exist a completion for $e_{ij} \in \mathcal{E}$ in $G = (\mathcal{V}, \mathcal{E})$?

```
while v_i, v_j ∈ V do
    p_ij ← Shortest path between v_i and v_j in G
    if ℓ(p_ij) + L_ij ≤ T_l for all v_l ∈ p_ij then
        return True
    V ← V \ {v_l ∈ V | ℓ(p_ij) + L_ij > T_l}
return False
```

Using the algorithm of Fredman and Tarjan [48], a shortest path in a weighted undirected graph can be computed in $\mathcal{O}(|E| + |V| \cdot \log|V|)$. Since in each iteration at least one vertex is removed, there are at most $|V| - 1$ iterations. Thus, Algorithm 9 has a run time of $\mathcal{O}(|V||E| + |V|^2 \cdot \log|V|)$. □

Using Algorithm 9, we can iterate over the edges in arbitrary order, delete the ones for which it returns `False`, and thereby remove all edges without a completion. To see this, suppose there is an edge $e_{ij} \in \mathcal{E}$ such that some other edge $e_{kl} \in \mathcal{E}$, which is part of a completion p_{ij} for e_{ij}, is removed in a subsequent iteration. Consider

the cycle induced by e_{ij} and p_{ij}. When removing e_{kl} from it, the resulting path p_{kl} is a completion for e_{kl}, since

$$\ell(p_{ij}) + L_{ij} = \ell(p_{kl}) + L_{kl} \leq T_s \text{ for all } v_s \in p_{kl},$$

which is a contradiction. Hence, for all remaining edges, there exists some completion and, therefore, a feasible cycle containing it.

5.4. Conflict Hypergraphs for LCCP

In this section, we introduce the concept of conflict hypergraphs for LCCP. These graphs contain information about subsets of vertices that cannot be contained in common feasible cycles. In particular, we are interested in the cliques of these graphs since they give rise to constraints that can be imposed on the vertices when modeling LCCP as a mathematical program.

Definition 8. *A hypergraph is a pair $H = (\mathcal{V}, \mathcal{E}^H)$ with vertex set $\mathcal{V}$ and hyperedge set $\mathcal{E}^H$. A hyperedge $e_S \in \mathcal{E}^H$ is a subset $S \subseteq \mathcal{V}$. H is called a c-uniform hypergraph or c-hypergraph if $|S| = c$ holds for all hyperedges $e_S \in \mathcal{E}^H_c$.*

Definition 9. *Let $H_c = (\mathcal{V}, \mathcal{E}^H_c)$ be a c-hypergraph. A hyperclique is a set $U \subseteq \mathcal{V}$ such that for each subset $S \subseteq U$ with $|S| = c$ we have $e_S \in \mathcal{E}^H$.*

Definition 10. *Let $(\mathcal{V}, \mathcal{E}, T, L)$ be an instance of LCCP. Its conflict c-hypergraph $H_c = (\mathcal{V}, \mathcal{E}^H_c)$ has $\mathcal{V}$ as vertex set and there is a hyperedge $e_S \in \mathcal{E}^H_c$ if no feasible cycle containing all vertices of $S \subseteq \mathcal{V}$ with $|S| = c$ exists.*

Corollary 9. *Let $(\mathcal{V}, \mathcal{E}, T, L)$ be an LCCP instance, and let $U \subseteq \mathcal{V}$ be a hyperclique in H_c. A feasible cycle C_k contains at most $c-1$ vertices from U, i.e., $|U \cap \mathcal{V}_k| \leq c-1$.*

Lemma 12. *Let $(\mathcal{V}, \mathcal{E}, T, L)$ be an LCCP instance and let $U \subseteq \mathcal{V}$ be a hyperclique of size $|U| = m$ in H_c. For each feasible cycle partition $\mathcal{C}$ we have $|\mathcal{C}| \geq \lceil \frac{m}{c-1} \rceil$.*

Proof. Let $\mathcal{C}$ be a cycle partition with $|\mathcal{C}| < \lceil \frac{m}{c-1} \rceil$. By the pigeonhole principle, there exists a cycle containing at least c vertices from U, which contradicts Corollary 9. □

Corollary 10. *Let $(\mathcal{V}, \mathcal{E}, T, L)$ be an LCCP instance and let $U \subseteq \mathcal{V}$ with $|U| = m$ be a maximum clique in H_c. Then $\lceil \frac{m}{c-1} \rceil$ is a lower bound on the size of an optimal cycle partition.*

Determining a shortest cycle that contains a given subset $S \subseteq \mathcal{V}$ of vertices is an NP-hard problem since TSP can be reduced to it. However, for pairs of vertices, i.e., if $|S| = 2$, it can be done in polynomial time.

Lemma 13. *Let $(\mathcal{V}, \mathcal{E}, T, L)$ be an LCCP instance, and let $v_i, v_j \in \mathcal{V}$. A shortest feasible cycle containing both vertices can be determined in $\mathcal{O}(|V||E| + |V|^2 \cdot \log |V|)$.*

Proof. Each feasible cycle containing v_i and v_j can be split into two vertex disjoint, except for the start- and end-vertex, paths p_{ij}^1 and p_{ij}^2. Hence, we can equivalently determine two such paths that minimize the expression $\ell(p_{ij}^1) + \ell(p_{ij}^2)$ to determine a smallest feasible cycle containing both vertices. This can be done with a modified version of Suurballe's algorithm [156], see Algorithm 10.

Algorithm 10: Does there exist a feasible cycle in $G = (\mathcal{V}, \mathcal{E})$ containing $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$?

```
while v_i, v_j ∈ V do
    p_ij^1, p_ij^2 ← Two vertex disjoint v_i-v_j-paths in G minimizing ℓ(p_ij^1) + ℓ(p_ij^2)
    if ℓ(p_ij^1) + ℓ(p_ij^2) ≤ T_l for all v_l ∈ p_ij^1 ∪ p_ij^2 then
        return True
    V ← V \ {v_l ∈ G | ℓ(p_ij^1) + ℓ(p_ij^2) > T_l}
return False
```

In each iteration, two vertex disjoint shortest paths p_{ij}^1 and p_{ij}^2 w.r.t. to the sum of their lengths are determined. We check if the critical weights of the vertices contained in both paths are respected. If this is the case, p_{ij}^1 and p_{ij}^2 induce a shortest feasible cycle containing v_i and v_j. Otherwise, we remove all vertices from G whose critical weights are smaller than $\ell(p_{ij}^1) + \ell(p_{ij}^2)$ and continue with the next iteration. Since the removal of vertices does not decrease the sum of the lengths of two vertex disjoint paths, the removed vertices cannot be part of any feasible cycle containing v_i and v_j. Moreover, since at least one vertex is removed after each iteration, there are at most $|V| - 1$ iterations. If v_i or v_j is removed at some point, there exists no feasible cycle containing both vertices and the algorithm terminates. Determining two vertex disjoint paths in a weighted undirected graph can be done in $\mathcal{O}(|E| + |V| \cdot \log |V|)$ using Suurballe's algorithm. Hence, the total running time of Algorithm 10 is $\mathcal{O}(|V||E| + |V|^2 \cdot \log |V|)$. □

By applying Algorithm 10 to each vertex pair $S := \{v_i, v_j\} \subseteq \mathcal{V}$, we can check whether $e_S \in \mathcal{E}_2^H$ or not and create the complete conflict (hyper)graph H_2.

Remark 2. *After an iteration of Algorithm 10, we may delete all vertices contained in* $\{v_l \in G \,|\, \ell(p_{ij}^1) + \ell(p_{ij}^2) > T_l\}$ *from* G. *This implies that there does not exist a feasible cycle containing* $S := \{v_i, v_j, v_l\} \subseteq \mathcal{V}$ *for each deleted vertex* v_l. *Hence, the corresponding hyperedges* e_S *are contained in* H_3.

5.4.1. Relation to Vertex Coloring

LCCP has a close connection to the Vertex Coloring Problem on its conflict hypergraphs. Recall the definition of the chromatic number.

Definition 11. *Let* H *be a hypergraph. The chromatic number* $\chi(H)$ *is the minimum number of colors needed to color the vertices such that no hyperedge is monochromatic, i.e., each hyperedge contains vertices of at least two colors.*

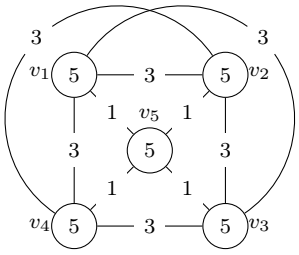


Figure 5.2.: Non-metric LCCP instance for which Lemma 15 does not hold.

Lemma 14. *Let $(\mathcal{V}, \mathcal{E}, T, L)$ be an LCCP instance. Further, consider the hypergraph $H := (V, \bigcup_{r=2}^{n} \mathcal{E}_r^H)$, i.e., the union of its c-conflict graphs. Then $\chi(H)$ is a lower bound on the size of an optimal cycle partition.*

Proof. The cycles of an optimal cycle partition are feasible by definition. Hence, no subset of vertices contained in a common cycle forms a hyperedge in H. Hence, assigning the same color to all vertices in a cycle yields a feasible coloring for H. □

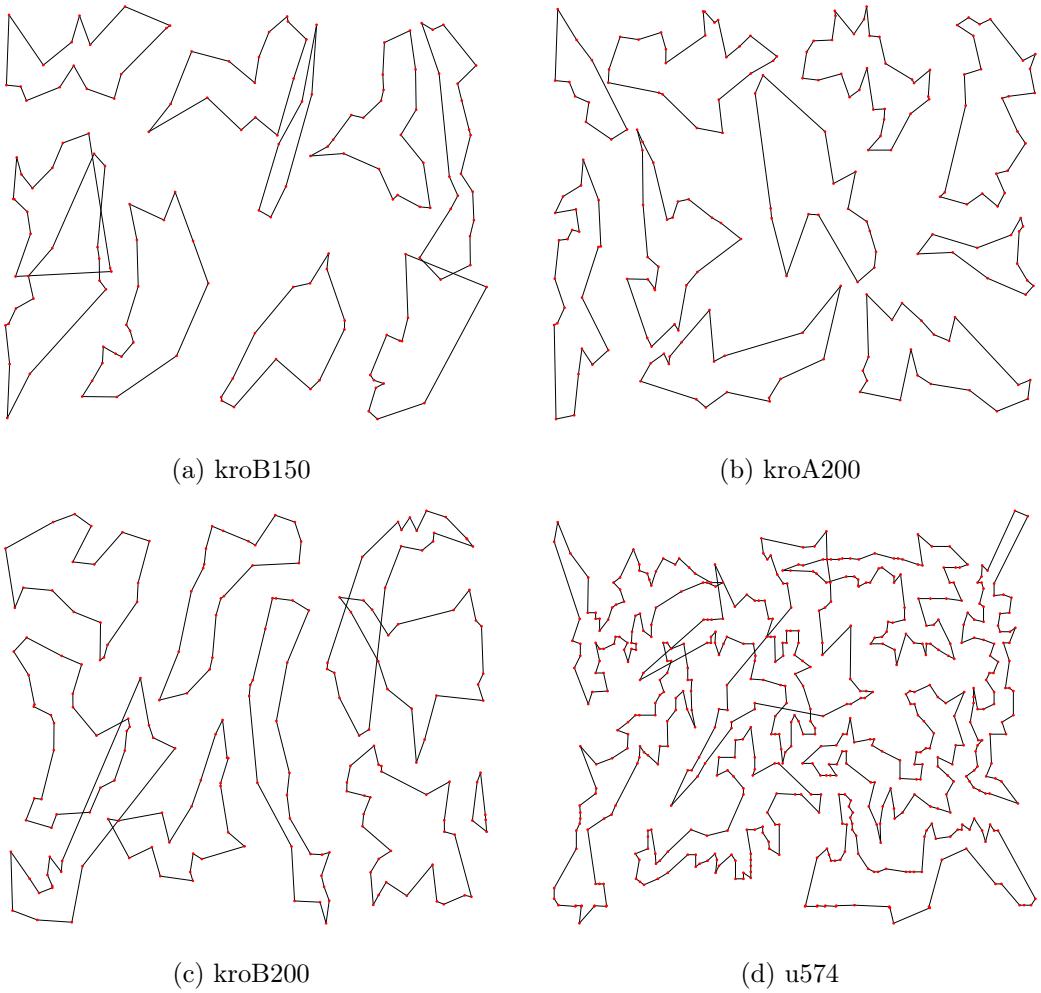
Corollary 11. *Let $(\mathcal{V}, \mathcal{E}, T, L)$ be an LCCP instance and let H_c be its corresponding conflict c-hypergraph. Then $\chi(H_c)$ is a lower bound on the size of an optimal cycle partition.*

For metric LCCP, we can even prove the following correspondence lemma.

Lemma 15. *Let $(\mathcal{V}, \mathcal{E}, T, L)$ be a metric LCCP instance. Consider the hypergraph $H := (V, \bigcup_{r=2}^{n} \mathcal{E}_r^H)$. The size of a minimum cycle partition is equal to the chromatic number of H. In particular, there exists a one-to-one mapping of feasible cycle partitions and feasible colorings of H.*

Proof. Given a feasible coloring of H, there exists a feasible cycle C containing all vertices of the same color. W.l.o.g. we can assume that this cycle C does not contain vertices of any other colour, since we can remove them due to the triangle inequality. Therefore, the colors induce a feasible cycle partition which has the same size as the coloring. On the other hand, given a feasible cycle partition, assigning all the vertices of a feasible cycle the same color yields a feasible coloring by definition. □

Lemma 15 does not hold for non-metric LCCP instances as the example in Figure 5.2 demonstrates. Here, H_2 has no edges, since for any two vertices v_i, v_j with $i, j \in \{1, 2, 3, 4\}$ and $i < j$ we have that the set $\{v_i, v_j, v_5\}$ induces a feasible cycle. Accordingly, for H_3 we have $\mathcal{E}_3^H = \{\{v_1, v_2, v_3\}, \{v_1, v_2, v_4\}, \{v_1, v_3, v_4\}, \{v_2, v_3, v_4\}\}$, since all cycles containing three vertices from $\mathcal{V} \setminus \{v_5\}$ contain at least two edges with weight 3. For the same reason, H_4 and H_5 are complete. Now, assigning v_1 and v_2, as well as v_3, v_4, and v_5 common colors yields a feasible coloring for H and we have $\chi(H) = 2$. However, a minimum cycle partition is given by $\mathcal{C} = \{(\{v_1\}, \emptyset), (\{v_2\}, \emptyset), (\{v_3, v_4, v_5\}, (\{v_3, v_4\}, \{v_4, v_5\}, \{v_5, v_3\}))\}$ and $|\mathcal{C}| = 3$.



(a) kroB150 (b) kroA200

(c) kroB200 (d) u574

Figure 5.3.: Feasible solutions for LCCP instances found by our MCV heuristic during computational experiments, see Table C.6. While solutions (a)-(c) feature ten cycles, (d) shows nine cycles.

5.5. A Most-Critical-Vertex-Based Heuristic for LCCP

Next, we present the Most-Critical-Vertex-Based Heuristic (MCV) for LCCP. Here, cycles are created sequentially. Its basic version is stated in Algorithm 11. The key idea of MCV is the following: The creation of a new cycle starts with a vertex having minimum critical weight among all the other vertices that have not yet been included in any other cycle. Next, this new cycle is extended by iteratively adding vertices to it, while its length does not exceed the critical weight value of the first vertex. The fact that the critical weights of the newly added vertices can be ignored in the process of extending the cycles is a notable advantage. It explains MCV's efficiency, which we will see in our computational experiments in Section 5.7. Note that although we apply the cheapest insertion heuristic [139] for extending the cycles in Algorithm 11 here, other appropriate TSP heuristics could also be adapted.

In the following, we explain Algorithm 11 in detail. $\tilde{\mathcal{V}} \subseteq \mathcal{V}$ denotes the set of vertices which have not yet been included in any cycle. In each iteration, a vertex $v_x \in \tilde{\mathcal{V}}$ with minimum critical weight in $\tilde{\mathcal{V}}$ is selected. If v_x is the only vertex in $\tilde{\mathcal{V}}$ or if there exists no adjacent vertex with which it forms a feasible cycle, it is returned as singleton, see lines 10 to 14. Otherwise, we continue with a shortest feasible cycle consisting of two vertices featuring v_x. In the following, we repeatedly determine two vertices v_a and v_b, which are adjacent in C_k, and a vertex $v_c \in \tilde{\mathcal{V}} \setminus \mathcal{V}_k$ such that the insertion of v_c between v_a and v_b would yield a minimum increase in the total cycle length. If the augmented cycle length does not exceed T_x, we insert v_c and continue to search for more suitable vertex triples, see lines 19 to 26. If we cannot extend C_k any further, we remove $\mathcal{V}_k$ from $\tilde{\mathcal{V}}$ and continue with the construction of the next cycle. The algorithm terminates when $\tilde{\mathcal{V}}$ is empty and each vertex is contained in a cycle. The final set of cycles is a feasible solution for LCCP.

This basic algorithm can be refined in line 24. Instead of returning C_k directly, any exact or heuristic algorithm for TSP can be applied to the subgraph of G induced by $\mathcal{V}_k$. If a tour of smaller length is found, i.e., τ_k can be decreased, further vertices could be inserted. In that case, the algorithm restarts the while-loop in line 19.

Moreover, the following observation motivates the introduction of an additional postprocessing routine for MCV: When Algorithm 11 terminates, cycle C_k, which was created last, often has small length and contains only a small number of vertices with large critical weights. Thus, the idea is to try to extend C_k by using vertices that are contained in the other cycles. If no such extension is possible, we return the current solution. Otherwise, we extend C_k in a similar manner as in lines 19 to 26, but in contrast, we dynamically adjust the minimum critical weight value of the cycle after each insertion and additionally check whether $\tau_k + \Delta\tau \leq T_c$ or not when determining a suitable vertex v_c. After extending C_k, we rerun Algorithm 11 with $\tilde{\mathcal{V}} := V \setminus V_k$. The described procedure is repeated until $\tilde{\mathcal{V}} = \emptyset$.

Algorithm 11 is of polynomial complexity. As demonstrated in our numerical experiments, MCV can successfully be used for both, producing an upper bound on the minimum number of cycles and approximately solving large instances of LCCP w.r.t. the number of vertices. The latter is of particular interest, since large instances

Algorithm 11: Most-Critical-Vertex-Based Heuristic (MCV) for LCCP

$\tilde{\mathcal{V}} \leftarrow \mathcal{V}$
$k \leftarrow 0$
while $\tilde{\mathcal{V}} \neq \emptyset$ **do**
  $k \leftarrow k + 1$
  $C_k \leftarrow$ `CreateFeasibleCycle`$(\tilde{\mathcal{V}}, \mathcal{E}, T, L)$
  $\tilde{\mathcal{V}} \leftarrow \tilde{\mathcal{V}} \setminus \mathcal{V}_k$
return $C_1, ..., C_k$

function `CreateFeasibleCycle`$(\tilde{\mathcal{V}}, \mathcal{E}, T, L)$
$v_x \leftarrow \arg\min_{v_i \in \tilde{\mathcal{V}}} T_i$
$N_x \leftarrow \{v_y \in \tilde{\mathcal{V}} \setminus \{v_x\} \,|\, e_{xy} \in \mathcal{E} \text{ and } T_x \geq 2 \cdot L_{xy}\}$
if $N_x = \emptyset$ **then**
  $C_k \leftarrow (\{v_x\}, \emptyset)$
  return C_k

$v_y \leftarrow \arg\min_{v_i \in N_x} L_{xi}$
$C_k \leftarrow (\{v_x, v_y\}, (\{v_x, v_y\}, \{v_y, v_x\}))$
$\tau_k \leftarrow 2 \cdot L_{xy}$
while $\tilde{\mathcal{V}} \setminus \mathcal{V}_k \neq \emptyset$ **do**
  $(v_a, v_b, v_c) \leftarrow$ vertices $v_a \in \mathcal{V}_k$ and $v_b \in \mathcal{V}_k$ being adjacent in C_k
    and $v_c \in \tilde{\mathcal{V}} \setminus \mathcal{V}_k$ with $e_{ac}, e_{bc} \in \mathcal{E}$ minimizing
    $\Delta\tau := -L_{ab} + L_{ac} + L_{cb}$
  if $\tau_k + \Delta\tau > T_x$ **then**
    return C_k
  $C_k \leftarrow (\mathcal{V}_k \cup \{v_c\}, (\mathcal{E}_k \cup \{\{v_a, v_c\}, \{v_c, v_b\}\}) \setminus \{v_a, v_b\})$
  $\tau_k \leftarrow \tau_k + \Delta\tau$

are computationally intractable for the MIP models proposed in Section 5.6. Some example solutions for large size instances produced by MCV are shown in Figure 5.3.

From a theoretical point of view, it is interesting to study the worst-case performance of MCV. Although the following lemma follows from LCCP not being in APX, the following proof is constructive as it describes a class of concrete example instances.

Lemma 16. *Algorithm 11 for LCCP has no constant approximation ratio.*

Proof. Consider the metric LCCP instance on the complete graph $G = (\mathcal{V}, \mathcal{E})$ with $|V| = n = 2k^2$ vertices. Let us denote $V_1 := \{v_1, \dots, v_k\}$ and $V_2 := \mathcal{V} \setminus V_1$. For each $v_i \in V_1$, let $T_i = 2k$, and for each $v_i \in V_2$, let $T_i = 2k^2 - k$. Additionally, let $L_{ij} = 2$ for each edge in $e_{ij} \in V_1 \times V_1$, and let $L_{ij} = 1$ otherwise. An optimal solution consists of the two cycles: One containing all vertices in V_1 and the other containing all vertices in V_2. In both cycles, the vertices are ordered according to their indices. However, the MCV heuristic produces a solution with k cycles, each featuring one vertex from V_1 and $2k - 1$ vertices from V_2. Hence, Algorithm 11 for LCCP does not admit a constant approximation ratio. □

5.6. Two MIP Models for LCCP

In this section, we present two MIP models for LCCP, which are inspired by two formulations for TSP. For their definition, we consider the induced directed graph $G = (\mathcal{V}, \mathcal{A})$, whose arc set contains the two directed arcs $a_{ij}, a_{ji} \in \mathcal{A}$ for each edge $e_{ij} \in \mathcal{E}$. Both arcs are assigned the same weight as the corresponding edge. Before discussing the differences between the two models, we first describe the variables and constraints they have in common.

First, for each vertex $v_i \in \mathcal{V}$, we introduce a binary variable y_i indicating whether it forms a singleton or not. In addition, for each potential proper feasible cycle C_k, we introduce a binary variable u_k with $k \in \mathcal{K} := \{1, \dots, \lfloor \frac{n}{2} \rfloor\}$ indicating whether it contains any vertices or not. Next, there is a nonnegative continuous variable τ_k representing the length of C_k. Further, for each vertex $v_i \in \mathcal{V}$ and each potential proper cycle C_k, we introduce a binary variable z_i^k indicating whether $v_i \in \mathcal{V}_k$ or not and analogously for each arc $a_{ij} \in \mathcal{A}$ a binary variable x_{ij}^k indicating whether $a_{ij} \in \mathcal{E}_k$ or not.

The objective function (5.1) aims at minimizing the total number of cycles, i.e., the sum of singletons and proper cycles. Constraints (5.2) ensure that each vertex either forms a singleton or is assigned to proper cycle. If a vertex v_i is assigned to proper cycle C_k, then (5.3) accounts for that and v_i has to have an outgoing and an ingoing arc, which is ensured by constraints (5.4) and (5.5), respectively. Next, constraints (5.6) keep track of the cycle lengths, while constraints (5.7) ensure that the critical weight values of all vertices are respected. Here, M_k denotes the k-th biggest critical weight among all vertices.

$$\min \sum_{v_i \in \mathcal{V}} y_i + \sum_{k \in \mathcal{K}} u_k \tag{5.1}$$

s.t.

$$y_i + \sum_{k \in \mathcal{K}} z_i^k = 1 \qquad \forall v_i \in \mathcal{V} \tag{5.2}$$

$$z_i^k \leq u_k \qquad \forall v_i \in \mathcal{V}, \forall k \in \mathcal{K} \tag{5.3}$$

$$\sum_{a_{ij} \in N^+(v_i)} x_{ij}^k = z_i^k \qquad \forall v_i \in \mathcal{V}, \forall k \in \mathcal{K} \tag{5.4}$$

$$\sum_{a_{ji} \in N^-(v_i)} x_{ji}^k = z_i^k \qquad \forall v_i \in \mathcal{V}, \forall k \in \mathcal{K} \tag{5.5}$$

$$\sum_{a_{ij} \in A} L_{ij} x_{ij}^k = \tau_k \qquad \forall k \in \mathcal{K} \tag{5.6}$$

$$T_i + (M_k - T_i)(1 - z_i^k) \geq \tau_k \qquad \forall v_i \in \mathcal{V}, \forall k \in \mathcal{K} \tag{5.7}$$

$$u_k \in \{0,1\} \quad \forall k \in \mathcal{K} \tag{5.8}$$

$$y_i \in \{0,1\} \quad \forall v_i \in \mathcal{V} \tag{5.9}$$

$$z_i^k \in \{0,1\} \quad \forall v_i \in \mathcal{V}, \forall k \in \mathcal{K} \tag{5.10}$$

$$x_{ij}^k \in \{0,1\} \quad \forall a_{ij} \in \mathcal{A}, \forall k \in \mathcal{K} \tag{5.11}$$

$$\tau_k \in \mathbb{R}_{\geq 0} \quad \forall k \in \mathcal{K} \tag{5.12}$$

The formulation above ensures that each vertex is contained in exactly one cycle and that all critical weight conditions are satisfied. Nevertheless, as it often occurs when designing MIP formulations for problems related to TSP, we have to take care of possible subtours. Hence, we extend this basic formulation in two different ways: One follows the idea of Miller, Tucker, and Zemlin of assigning an order to the vertices in order to prohibit subtours, which has first been introduced in [82]. The second one uses a novel and modified version of the well-known subtour elimination constraints.

5.6.1. Subtour Elimination Constraints

Subtour elimination constraints for TSP ensure that between any two nonempty sets of vertices there are at least two arcs connecting them. However, in contrast to TSP, we do not know in advance which vertices form a common cycle in LCCP. Thus, we cannot directly apply the classic subtour elimination constraints and therefore introduce constraints

$$\sum_{\substack{v_i, v_j \in S_1: \\ a_{ij} \in A}} x_{ij}^k + \sum_{\substack{v_i, v_j \in S_2: \\ a_{ij} \in A}} x_{ij}^k \leq |S_1| + |S_2| - 2 \quad \begin{array}{l} \forall S_1, S_2 \subset V, S_1, S_2 \neq \emptyset \\ S_1 \cap S_2 = \emptyset, \forall k \in \mathcal{K} \end{array} \tag{5.13}$$

instead. Assume that a subset of the vertices assigned to the set representing $\mathcal{V}_k$ in the basic MIP formulation (5.1)–(5.12) forms two proper cycles C_k^1 and C_k^2. In that case, constraint (5.13) with $S_1 = \mathcal{V}_k^1$ and $S_2 = \mathcal{V}_k^2$ is violated, since

$$\sum_{\substack{v_i,v_j \in C_k^1: \\ a_{ij} \in A}} x_{ij}^k + \sum_{\substack{v_i,v_j \in C_k^2: \\ a_{ij} \in A}} x_{ij}^k = |\mathcal{V}_k^1| + |\mathcal{V}_k^2| = |S_1| + |S_2|.$$

Conversely, if the vertices form one proper cycle, no sets S_1 and S_2 exist such that the corresponding constraint is violated. Hence, the MIP consisting of (5.1)–(5.12) and constraints (5.13), which we call SEC in the following, models LCCP. Although there can be exponentially many subtour elimination constraints, a potential way to algorithmically incorporate them is described in Subsection 5.7.3.

5.6.2. Miller–Tucker–Zemlin Formulation

An alternative way to avoid subtours in TSP is the Miller–Tucker–Zemlin (MTZ) formulation [112]. Here, each vertex is assigned a positive weight while the starting vertex has value zero. For each pair of consecutive vertices in a tour, the weights must increase except for the last and the starting vertex. Again, a straightforward use for LCCP is not possible, since we cannot fix the starting vertices for the cycles in advance. Thus, for each $k \in \mathcal{K}$ and each vertex $v_i \in \mathcal{V}$ we introduce additional binary variables $s_i^k \in \{0,1\}$ indicating whether v_i is the starting vertex of cycle C_k or not. Weight variables $w_i^k \in \mathbb{Z}_{\geq 0}$ together with constraints

$$\sum_{i \in \mathcal{V}} s_i^k = u_k \qquad \forall k \in \mathcal{K} \tag{5.14}$$

$$s_i^k \leq z_i^k \qquad \forall v_i \in \mathcal{V}, \forall k \in \mathcal{K} \tag{5.15}$$

$$\sum_{v_i \in \mathcal{V}} z_i^k - u_k \geq w_i^k \qquad \forall v_i \in \mathcal{V}, \forall k \in \mathcal{K} \tag{5.16}$$

$$w_i^k - w_j^k + |V| \cdot (x_{ij}^k - s_j^k) \leq |V| - 1 \qquad \forall a_{ij} \in A, \forall k \in \mathcal{K} \tag{5.17}$$

$$s_i^k \in \{0,1\} \qquad \forall v_i \in \mathcal{V}, \forall k \in \mathcal{K} \tag{5.18}$$

$$w_i^k \in \mathbb{Z}_{\geq 0} \qquad \forall v_i \in \mathcal{V}, \forall k \in \mathcal{K}. \tag{5.19}$$

then model the idea explained above applied to LCCP. Constraints (5.14) determine a starting vertex for each cycle, which also has to be part of it due to constraint (5.15). Furthermore, the necessary weight values are bounded by (5.16) for each set $k \in \mathcal{K}$. Finally, constraints (5.17) are the Miller–Tucker–Zemlin constraints as explained above. Thus, the MIP model consisting of (5.1)–(5.12) together with (5.14)–(5.19) is a model for LCCP and we denote it by MTZ in the following.

5.6.3. Symmetry Breaking Inequalities

The solution space of the two MIP models is highly symmetric. Given a feasible solution, all permutations of the proper cycle indices respecting constraints (5.7) are also feasible. Assume w.l.o.g. that the vertices are ordered non-increasingly by their critical weights. Then, inequalities

$$z_i^k \leq \sum_{j=1}^{i-1} z_j^{k-1} \quad \forall v_i \in \mathcal{V}, \forall k \in \mathcal{K} \setminus \{1\} \tag{5.20}$$

ensure that only the permutation with the cycles sorted non-decreasingly by the minimum index of the contained vertices remains feasible.

5.6.4. Conflict Clique Inequalities

Let $(\mathcal{V}, \mathcal{E}, T, L)$ be an LCCP instance and let H_c be its conflict c-hypergraph. Further, let $U \subseteq \mathcal{V}$ with $|U| = m$ be a hyperclique in H_c. From Corollary 9 we derive

$$\sum_{v_i \in U} z_i^k \leq c - 1 \quad \forall k \in \mathcal{K} \tag{5.21}$$

as valid inequalities for LCCP. In addition, the size of each clique induces a lower bound on the size of an optimal solution by Corollary 10. Thus,

$$\sum_{v_i \in \mathcal{V}} y_i + \sum_{k \in \mathcal{K}} u_k \geq \lceil \frac{m}{c-1} \rceil \tag{5.22}$$

is a valid inequality, too. The following lemma demonstrates, that none of the conflict hypergraphs is redundant w.r.t. conflict clique constraints.

Lemma 17. *Let $c \in \mathbb{N}$ with $c \geq 2$. There exists an LCCP instance $(\mathcal{V}, \mathcal{E}, T, L)$ and a feasible solution for the LP-relaxation of the corresponding MIP models, which is cut off by conflict clique constraints that can only be derived from the conflict c-hypergraph. Further, the lower bound induced by the size of a maximum hyperclique of this c-hypergraph is the only tight one, i.e., it is the only one equal to the size of an optimal solution.*

Proof. Consider the LCCP instance featuring the complete graph on $2c-1$ vertices where $T_i = 2c-1$ for each vertex $v_i \in \mathcal{V}$ and $L_{ij} = 2$ for each edge $e_{ij} \in \mathcal{E}$. For each subset $S \subseteq \mathcal{V}$ with $|S| < c$, there exist a feasible cycle, e.g., the cycle induced by any permutation of the vertices in S and the corresponding connecting edges. Hence, all conflict d-hypergraphs with $d < c$ are empty, i.e., they do not contain any hyperedge. On the other hand, if $|S| \geq c$, there does not exist any feasible cycle and all conflict d-hypergraphs with $d \geq c$ are complete.

A feasible solution for the LP-relaxations of the two MIP models is given by $u_k = \frac{1}{2}$, $z_j^k = \frac{1}{2}$, $x_{ij}^k = \frac{1}{2}$ for all e_{ij} with $j = i+1$ together with $e_{2c-1,1}$, and $\tau_k = 2c-1$

for $k \in \{1, 2\}$, while all remaining variables are zero. For model MTZ we additionally have $s_1^1 = s_1^2 = \frac{1}{2}$. Since the conflict d-hypergraphs with $d \geq c$ are complete, $\mathcal{V}$ is the maximum hyperclique in each of them. Therefore, the corresponding conflict clique constraints (5.21) for $k \in \{1, 2\}$, i.e.,

$$\sum_{v_i \in \mathcal{V}} z_i^k = \frac{1}{2}|\mathcal{V}| = c - \frac{1}{2} \leq d - 1,$$

are violated if and only if $d = c$. Furthermore, the lower bound (5.22) on the size of a smallest cycle partition is given by

$$\lceil \frac{2c-1}{c-1} \rceil \geq \frac{2c-1}{c-1} > \frac{2c-2}{c-1} = 2$$

for the conflict c-hypergraph, while for $d > c$ we have

$$\lceil \frac{2c-1}{d-1} \rceil \leq \lceil \frac{2c-1}{c} \rceil = \lceil 2 - \frac{1}{c} \rceil = 2.$$

Since an optimal cycle partition consists of three cycles, e.g., any cycle induced by the vertex sets $\mathcal{V}_1 := \{v_1, \ldots, v_{c-1}\}$ and $\mathcal{V}_2 := \{v_c, \ldots, v_{2c-2}\}$, and the singleton $\mathcal{V}_3 := \{v_{2c-1}\}$, we know that the bound for $d = c$ is tight. □

5.7. Computational Experiments

In this section, we describe, present, and analyze the computational experiments, which we conducted in order to test the MCV heuristic and both MIP approaches. Furthermore, we ran LCCP on publicly available CR-UAV instances [39, 40] and discuss a possible integration of it into existing CR-UAV algorithms.

5.7.1. Computational Setup

We ran all of our experiments on a cluster of machines composed of Intel Xeon Gold 5122 @ 3.60GHz CPUs with 96GB of RAM. All algorithms were implemented in Python and we used the corresponding interface of the LP-based branch-and-bound solver Gurobi v9.0 [68] with default parameters and a time limit of 6 hours.

5.7.2. Instances

For our computational experiments, we generated test instances based on the 28 instances from the TSPLIB and 14 instances from the ATSPLIB with 100 or fewer vertices [136, 135]. The original ATSPLIB instances are defined on a directed graph and the corresponding arc weights are given using an asymmetric matrix. To create our LCCP instances, we chose the weights in the upper triangular part.

Furthermore, let τ^* denote the length of an optimal tour for the corresponding TSP or ATSP instance. We created two test sets based on these instances. For the

first test set, we assigned each vertex a random integer from the interval $[\frac{\tau^*}{6}, \frac{\tau^*}{2}]$ as critical weight. For the second test set, we assigned an integer from $[\frac{\tau^*}{8}, \frac{\tau^*}{4}]$.

5.7.3. Algorithmic Setup

First of all, we removed all unusable edges using Algorithm 9.

Next, we computed a feasible solution using the MCV heuristic. Thereby, we simultaneously derived an upper bound on the necessary size of the set $\mathcal{K}$, which we use in our MIP models. In particular, we applied the refinement step of MCV, where we used the exact MIP formulation of Dantzig, Fulkerson and Johnson [29] for solving the induced TSP to reduce the cycle length. Moreover, we also used the proposed postprocessing routine.

Afterward, we determined the complete conflict 2-hypergraph using Algorithm 10 and determined all maximal cliques using the algorithm of Cazals and Karande [22]. The corresponding constraints (5.21), the lower bound (5.22), and the symmetry breaking constraints (5.20) were added to both MIP models.

Further, we determined a subset $\tilde{\mathcal{E}}_3^H \subseteq \mathcal{E}_3^H$ of the hyperedges of H_3 as follows. For each vertex triple $v_i, v_j, v_l \in \mathcal{V}$, we checked whether the sum of the shortest paths between the three vertices is larger than $\min\{T_i, T_j, T_l\}$. Additionally, we added all the edges obtained according to Remark 2. All mentioned calculations up to this point were performed in less than 2 minutes for each test instance.

While the corresponding variables and constraints for the MTZ model were added before the start of the solving process, the subtour elimination constraints (5.13) for SEC were separated during the solving process. Let $\mathcal{C} = \{C_1, \dots, C_m\}$ be a solution for the current formulation. If C_k contains subtours, we add the corresponding SEC constraint (5.13) for every pair of distinct subtours and each $k \in \mathcal{K}$.

Moreover, we also heuristically separated constraints from the cliques of our subset of hyperedges of H_3 for both models during the solving process. Therefore, after solving the LP relaxation of each branch-and-bound node, we determined hypercliques in $\tilde{H}_3 := (\mathcal{V}, \tilde{\mathcal{E}}_3^H)$ using a greedy routine: Let $\tilde{z}$ denote the vector of z-variable values in the current LP-solution, consider the subgraph induced by the vertex set $\{v_i \in \mathcal{V} \mid \tilde{z}_i^k > 0\}$ for some $k \in \mathcal{K}$. Using a greedy routine on the hyperedges w.r.t. $\tilde{z}$, we compute a maximal clique in this subgraph. Afterward, we extend this clique to a maximal clique U in $\tilde{H}_3$ by adding suitable vertices with $\tilde{z}_i^k = 0$. The corresponding constraint (5.21) is then added to the model and cuts off the current LP-solution if $\sum_{v_i \in U} \tilde{z}_i^k > 2$.

5.7.4. Computational Results

In the following analysis, we call instances with less than 29 vertices small, less than 59 vertices medium-sized, and all others are referred to as large. While all small instances were solved within a few seconds, all large instances ran into the time limit. Thus, we focus on the results for medium-sized instances in the following, which are shown in Tables 5.1 to 5.4. The detailed results for all instances can be

Table 5.1.: Results for TSPLIB with critical weights from $[\frac{\tau^*}{8}, \frac{\tau^*}{4}]$

Instance name	Removed edges in %	UB (MCV heur)	LB (H_2-clique)	MTZ UB	MTZ LB	MTZ time (sec)	SEC UB	SEC LB	SEC time (sec)
bayg29	58.6	10	7	8	8	42	8	8	23
bays29	58.1	8	6	8	8	36	8	8	21
dantzig42	56.1	9	7	9	9	1552	9	9	575
swiss42	52.4	9	7	9	8	TL	9	9	284
att48	53.6	9	6	8	8	4563	8	8	2893
gr48	48.0	9	5	9	8	TL	9	8	TL
hk48	48.6	10	6	9	8	TL	9	9	14417
eil51	40.9	9	5	9	8	TL	9	9	19242
berlin52	38.8	9	6	9	9	2956	9	9	2260
brazil58	40.9	9	5	8	7	TL	9	7	TL

Table 5.2.: Results for TSPLIB with critical weights from $[\frac{\tau^*}{6}, \frac{\tau^*}{2}]$

Instance name	Removed edges in %	UB (MCV heur)	LB (H_2-clique)	MTZ UB	MTZ LB	MTZ time (sec)	SEC UB	SEC LB	SEC time (sec)
bayg29	25.4	6	4	5	5	30	5	5	13
bays29	27.6	6	5	6	6	32	6	6	14
dantzig42	28.7	7	4	6	6	20686	6	6	15081
swiss42	18.6	7	4	6	6	13111	6	6	6833
att48	26.5	8	5	6	5	TL	6	6	4939
gr48	18.5	7	4	7	5	TL	7	6	TL
hk48	25.0	7	4	6	6	18843	6	6	3656
eil51	10.0	7	3	7	5	TL	7	5	TL
berlin52	19.9	7	4	7	6	TL	7	6	TL
brazil58	14.7	6	3	6	4	TL	6	5	TL

found in Tables C.1 to C.4 in Appendix C.1. The instance names in the first columns incorporate the number of vertices, i.e., instance bayg29 features 29 vertices. For ATSPLIB instances with the "ftv"-prefix, we must add one vertex, e.g., ftv33 has 34 vertices. Instance kro124p is an outlier w.r.t. this nomenclature and features 100 vertices. In the second column, the percentage of edges that were removed during preprocessing is shown. Furthermore, while the value of the solution found by the MCV heuristic is stated in column UB (MCV heur), the lower bound obtained from the size of a maximum clique in the conflict graph H_2 is shown in column LB (H_2-clique). For both MIP approaches, the upper and the lower bound at the end of the solving processes are stated, which was either reached when the problem was solved or when the time limit was hit, which is indicated by TL in the time column.

The results for the TSPLIB-based test sets can be found in Table 5.1 for critical weights in $[\frac{\tau^*}{8}, \frac{\tau^*}{4}]$ and Table 5.2 for critical weights in $[\frac{\tau^*}{6}, \frac{\tau^*}{2}]$. For the first test set,

Table 5.3.: Results ATSPLIB with critical weights from $[\frac{\tau^*}{8}, \frac{\tau^*}{4}]$

Instance name	Removed edges in %	UB (MCV heur)	LB (H_2-clique)	MTZ UB	MTZ LB	MTZ time (sec)	SEC UB	SEC LB	SEC time (sec)
ftv33	57.9	9	7	8	8	120	8	8	137
ftv35	53.3	9	6	8	8	205	8	8	149
ftv38	48.2	10	6	9	8	TL	9	9	2614
p43	25.0	4	2	4	3	TL	3	3	1398
ftv44	45.6	9	6	9	8	TL	9	9	3297
ftv47	40.6	11	5	11	8	TL	11	8	TL
ry48p	40.2	9	5	9	8	TL	9	8	TL
ft53	10.7	11	3	11	6	TL	11	6	TL
ftv55	40.3	10	5	10	7	TL	10	7	TL

we can solve all but two medium-sized instances. Thereby, model SEC can solve three more instances than model MTZ and is faster on all medium-sized instances. The results of the second test set are similar: SEC can solve one more instance than MTZ and is faster on all instances. However, compared to the first test set, it takes longer to solve the instances, and four were not solved at all. This is probably due to the larger critical weights, which allow for a higher degree of freedom when creating feasible cycles.

The results for the ATSPLIB-based test sets can be found in Table 5.3 for critical weights in $[\frac{\tau^*}{8}, \frac{\tau^*}{4}]$ and Table 5.4 for critical weights in $[\frac{\tau^*}{6}, \frac{\tau^*}{2}]$. The results in Table 5.3 are similar to the ones for the corresponding metric test instance set in Table 5.1. We can solve instances with up to 44 vertices using model SEC, while model MTZ could prove optimality only for two medium-sized instances. However, for critical weights in $[\frac{\tau^*}{6}, \frac{\tau^*}{2}]$, model SEC solves only one instance, and model MTZ solved two instances, see Table 5.4. Again, due to bigger critical weights, there is a higher degree of freedom when it comes to the creation of feasible cycles. Thus, these critical weights combined with general edge weights apparently make the problem more challenging.

Since it provided high-quality solutions and ran for less than one second on all instances considered so far, we wanted to analyze the performance of the MCV heuristic on larger instances in more depth. Therefore, we created additional test instances, which we derived by applying the same procedure as described above to all TSPLIB and ATSPLIB instances with more than 100 and less than or equal to 1000 vertices. Again, the instance names incorporate the number of vertices in the same way as stated above.

We experimented with two variants of MCV: The first on is equivalent to the variant described above. We again applied the exact TSP algorithm and tried to decrease the length of each cycle after its creation - this version we denote by MCV Enhanced. For the second variant, we omit this step and denote this approach by MCV Basic. The results can be found in Tables C.5 to C.8 in Appendix C.1.

Table 5.4.: Results ATSPLIB with critical weights from $[\frac{\tau^*}{6}, \frac{\tau^*}{2}]$

Instance name	Removed edges in %	UB (MCV heur)	LB (H_2-clique)	MTZ UB	MTZ LB	MTZ time (sec)	SEC UB	SEC LB	SEC time (sec)
ftv33	23.9	7	4	7	6	TL	7	6	TL
ftv35	20.0	6	3	5	5	14852	6	5	TL
ftv38	11.7	7	3	7	5	TL	6	5	TL
p43	22.0	3	2	3	3	1	3	3	1
ftv44	11.6	8	3	8	5	TL	8	6	TL
ftv47	14.8	7	3	7	5	TL	7	6	TL
ry48p	14.0	7	3	7	5	TL	7	5	TL
ft53	0.6	7	2	7	4	TL	7	4	TL
ftv55	11.8	8	4	8	5	TL	8	5	TL

Both heuristic variants perform well, as the size of nearly all obtained solutions features around ten cycles. Thereby, MCV Enhanced has a slight advantage and often reduces the size of the solutions by one or two cycles compared to MCV Basic. However, we observe that the run times of both approaches increase with the size of the instances. Thereby, the increase for MCV Enhanced is stronger and goes up to around nearly 1000 seconds. The reason is that the TSP instances solved in between are becoming more challenging and take up more time. In contrast, the maximum run time of MCV Basic is about 80 seconds. Thus, the results obtained here demonstrate that MCV is, in general, well suited for finding solutions of good quality in short amounts of time, even for large instances.

5.7.5. Additional Experiments Regarding CR-UAV

As discussed in Section 5.2, LCCP is closely related to CR-UAV as each feasible solution for LCCP represents a feasible solution for CR-UAV. However, the reverse statement does not hold, which directly follows from the following lemma.

Lemma 18. *The ratio of the optimal solution values for corresponding LCCP and CR-UAV instances is not bounded by a constant.*

Proof. Consider the complete graph $G = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n + 1$ vertices. While we define $T_1 = 2$ for $v_1 \in \mathcal{V}$, we set $T_i = 2n$ for all other vertices $v_i \in \mathcal{V} \setminus \{v_1\}$. Furthermore, let $L_{1j} = 1$ for each edge $e_{1j} \in \mathcal{E}$ that is incident to v_1, and let $L_{ij} = 2n$ hold for all other edges.

An optimal solution for the corresponding CR-UAV instance consists of one cyclic route, e.g., $(v_1, e_{12}, v_2, e_{12}, v_1, e_{13}, v_3, e_{13}, v_1, \ldots, v_1, e_{1n+1}, v_{n+1}, e_{1n+1}, v_1)$. On the other hand, an optimal solution for the corresponding LCCP instance consists of n cycles, e.g., the cycle connecting v_1 and v_2 and all other vertices being singletons. □

Moreover, we ran the SEC model for LCCP on the 300 test instances presented by Drucker et al. [40], which are publicly available at [39]. The underlying graphs

Satisfiable+	Satisfiable	Not Satisfiable	No Statement Possible
103	76	20	101

Table 5.5.: Results of the experiments comparing the optimal solution values for LCCP with the fixed number of UAVs used in the corresponding satisfiability problems from [39]. Satisfiable+ denotes the number of instances where LCCP determined a solution with fewer UAVs, while Satisfiable relates to instances where their number is equal. For 20 instances, we show that they are not satisfiable.

feature up to 7 vertices, and all of them were solved in less than 0.1 seconds in the root nodes of the branch-and-bound trees. Hence, since LCCP yields an upper bound on the number of UAVs, it is well suited to be applied within solution algorithms for CR-UAV as it comes with no computational expense regarding run time.

Motivated by Lemma 18, we investigated the quality of our solutions, too. The instances from [39] feature a fixed parameter representing the number of UAVs since they correspond to satisfiability problems solved within the solution approach of Drucker et al. [40], see Section 5.2. Hence, we compared the optimal LCCP solution values with this parameter, which was either one, two, or three. The results of these experiments are summarized in Table 5.5.

For 103 instances, we found a feasible solution implying that even fewer UAVs are necessary, while for 76 of them, we obtained optimal solution values equal to the parameter. In other words, for about 60% of the instances, we can show that the instances are satisfiable. On the other hand, for 20 instances (6%), where only one UAV is available, we can show that they are not satisfiable because the corresponding optimal LCCP solution consists of singletons only. For all other instances, no statement can be made. Thus, comparing this performance with the average run times of 26.8 seconds presented in [40] is another argument for applying our solution approaches for LCCP within CR-UAV algorithms.

5.8. Conclusion and Future Research

Chapter 5 starts with a discussion on the idea of using uncrewed aerial vehicles (UAVs) to monitor the gas transport infrastructure. Thereby, the necessity of regularly checking on the network elements is motivated using the example of pipeline leakage. In this context, we introduce and discuss the Length-Constrained Cycle Partition Problem (LCCP), a routing problem regarding these UAVs in Section 5.1, representing a new generalization of the Traveling Salesperson Problem (TSP).

As its name indicates, LCCP is about partitioning an undirected graph into vertex-disjoint cycles. These cycles represent flying routes for the UAVs, and we require them to be vertex-disjoint to avoid interferences. Further, each vertex, which corresponds to an area or an object that shall be monitored, is assigned a critical weight

value, which represents the maximum duration for which it can be left unattended after its last scan. We require that the length of each cycle is not greater than a smallest critical weight value of a vertex contained in it. The goal of LCCP is to find a feasible partition with the smallest number of cycles, i.e., the smallest number of UAVs necessary to monitor all areas.

Besides the Most-Critical-Vertex-Based Heuristic (MCV) heuristic introduced in Section 5.5, which yields convincing results in our computational experiments even for large instances, we present two exact MIP formulations for LCCP in Section 5.6. While the first one incorporates an adaption of the MTZ model for TSP, the second one is based on a generalization of subtour elimination constraints. Symmetry-breaking constraints are introduced to reduce the number of feasible solutions. Moreover, we introduce the notion of conflict hypergraphs for LCCP in Section 5.4. Here, a hyperedge represents a set of vertices that cannot be contained in a common cycle. Based on the cliques of these graphs, we derive lower bounds and another class of valid inequalities for LCCP. Finally, we can determine optimal solutions for TSPLIB- and ATSPLIB-based problem instances with up to fifty vertices, see Section 5.7.

There are several directions for future research. First of all, it remains an open question whether there exists an algorithm with constant approximation ratio for metric LCCP or not. Currently, we do not dare to state a conjecture here as both options seem equally likely to us. However, despite its convincing performance in our computational experiments, the fact that the MCV heuristic does not admit a constant approximation ratio can be seen as an indicator for the need for more elaborate algorithmic ideas. We believe that straightforward adaptions of classical TSP strategies will not be successful in this context.

From a computational perspective, we are currently working on identifying additional valid inequalities and are developing a problem-specific branching strategy to solve instances with even more vertices. In this context, we also aim to exploit the relation of LCCP's conflict hypergraphs to the corresponding vertex coloring problems, which is described in Section 5.4.1, to derive stronger lower bounds.

Finally, we are going to pay more attention to the Cyclic Routing of UAVs Problem (CR-UAV), which we described in Section 5.2. In this natural generalization of LCCP, the flying routes of the UAVs have to be cyclic, too. However, they are allowed to intersect, and the critical weight conditions can be satisfied jointly by specifying coordinated flying schedules. Lemma 18 demonstrated, there is quite some potential regarding the minimum number of necessary UAVs for some instance classes when considering CR-UAV instead of LCCP. However, CR-UAV is very challenging and all instances considered in the literature so far feature less than ten vertices. Further, the resulting solutions and corresponding flight schedules could become quite complex, which may be a disadvantage in practice compared to LCCP. Therefore, we will concentrate on heuristic algorithms, which produce solutions of good quality while maintaining a simple structure. In particular, we aim at establishing a middle ground here: While we insist that the UAVs continue to fly cycles, like in LCCP, their routes are allowed to intersect and the critical weight conditions can be satisfied jointly, as in CR-UAV.

6. General Conclusion

In this thesis, we discussed four real-world problems arising in the area of gas transport networks. They originate from different domains, i.e., optimizing transient network control, estimating a network's capacity by identifying worst-case transport scenarios, and efficiently monitoring the transport infrastructure using uncrewed aerial vehicles (UAVs). After defining and analyzing these problems, we presented modeling and solution approaches based on mathematical programming. We summarize our work as follows:

- ▷ We contributed to the development of the first decision support system for the transient control of natural gas transport networks: KOMPASS. In this context, we introduced a tri-level MIP model to determine important global control decisions, e.g., how to route the flow and where and when to compress the gas. To this end, we applied the modeling concept of network stations to approximate the technical control capabilities. A sequential linear programming inspired post-processing routine is run to derive physically accurate solutions w.r.t. the transient gas flow in pipelines. Computational experiments based on real-world data confirm the validity of our approach and its potential for the further development of KOMPASS towards a fully automated decision-making system. (Chapter 3)
- ▷ We adapted the aforementioned algorithmic approach to handle control problems regarding hydrogen networks and proposed a method for converting input data from natural gas transport into corresponding transport scenarios for hydrogen. Based on this, we conducted computational experiments to investigate whether the natural gas infrastructure can be repurposed for hydrogen transport and if and how this changes the network control. We find that the installed turbo compressors must be replaced with (multiple, parallel) special hydrogen compressors to transport energy-equivalent amounts of hydrogen. Further, due to the lower energy density of hydrogen, the linepack w.r.t. energy decreases significantly. This results in more technical control measures as the hydrogen must be stored in or retrieved from more remote parts of the network. Besides an average increase of 440% in compression energy compared to natural gas transport, we observe that supply and demand must be balanced in shorter time intervals than the currently established twenty-four hours to ensure a feasible technical network control. Hence, repurposing the infrastructure seems possible if the mentioned criteria are satisfied. However, network control becomes more complicated, which would make a decision support system such as KOMPASS even more valuable. (Chapter 3)

- ▷ We discussed the problem of identifying severe transport scenarios, which is a crucial task when determining the technical capacity of a gas network. In this context, we proposed two new severity measures: The minimum transport moment and the potential transport moment. To identify corresponding worst-case scenarios, we introduced the Maximum Transportation Problem (MaxTP) and the Maximum Potential Transport Moment Problem (MaxPTM). Both problems were modeled as linear bilevel programs with interdicting objective functions where the leader selects a transport scenario that maximizes the transport effort of the follower. In MaxTP, the latter is equivalent to the optimal solution value of the Transportation Problem induced by the corresponding supplies and demands, while a linear potential-based flow formulation is applied in MaxPTM. MaxPTM can be reformulated as MIP. For MaxTP, we constructed a solution-equivalent instance with a tripartite, acyclic network structure, derived bounds for the variables of the corresponding KKT reformulation, and applied the big-M technique to derive a MIP model. A case study on the gaslib-582 instance from the GasLib demonstrates that our severity measures and the corresponding algorithmic methods should be incorporated into existing routines for determining test sets of severe transport scenarios in order to improve their coverage. (Chapter 4)
- ▷ We discussed the usage of UAVs to monitor the transport infrastructure as well as their routing, which motivated the introduction of the Length-Constrained Cycle Partition Problem (LCCP). Its goal is to find a smallest cycle partition that satisfies additional vertex-induced length requirements. Besides our Most-Critical-Vertex-Based Heuristic, which yields convincing results in TSPLIB- and ATSPLIB-based computational experiments for large instances, we introduced two exact MIP formulations. While the first one incorporates an adaption of the MTZ model for TSP, the second one is based on a generalization of subtour elimination constraints. Combining them with valid inequalities and lower bounds derived from cliques of conflict hypergraphs as well as symmetry-breaking constraints, we determined optimal solutions for problem instances with up to fifty vertices. This number is significantly larger than the instance sizes that are currently tractable for similar, related optimization problems, e.g., CR-UAV with less than ten vertices. (Chapter 5)

For more detailed discussions and outlooks on possible directions for future research regarding the work presented in the three main chapters, we refer to Section 3.11, Section 4.5, and Section 5.8, respectively.

In recent years, various compelling energy-related optimization problems have been successfully solved by mathematical programming approaches. These complex applications often stimulate research regarding new algorithmic approaches and underlying combinatorial structures. At the same time, the technological and theoretical advancements in solving large-scale mathematical programs, especially LP and MIP, constantly enable and motivate researchers to tackle novel or previously intractable problems.

We strongly believe that there will continue to be a fruitful synergy between operations research in the area of gas transport networks and mathematical programming, and we consider our results as another small piece of this puzzle. In this spirit, we hope that our work will help to raise awareness for critical issues related to gas transport networks and their impact on society, and demonstrate the power of mathematical programming and its inherent beauty.

A. Appendix to Chapter 3

A.1. Modeling Parameters and Weights

This section states the parameters and weights used in the different algorithmic components of Algorithm 4 w.r.t. our computational experiments.

First, for the tri-level MIP model we considered a temporal granularity of $2 \cdot 60$ minutes and $11 \cdot 120$ minutes for all instances, i.e., we have $k = 13$ time steps covering twenty-four hours. The cost parameters regarding the third level objective were $w^a = 5$ for all $a \in \mathcal{A}^{\text{ar}}$, and for each simple state $s \in \mathcal{S}$ an individual cost $w^s \in \{0, \ldots, 200\}$ for a change into it was selected according to the practitioners. Note that the cost for both classes of slack variables is equal to 1, see Section 3.5.5 on the objective functions. Finally, as minimum threshold for the absolute velocity used within the Momentum Equations (3.5), we applied $v^{min} = 10^{-1} \frac{m}{s}$.

Second, in the solution smoothing LP, which is described in Subsection 3.6.2, for each network station G_i and each $v \in \mathcal{V}_i^{\text{fn}}$ we used $w_v^{\text{sm-q}} = 13$, $w_i^{\text{sm-q}} = |\mathcal{V}_i^{\text{fn}}|$, $w_v^{\text{sm-p}} = 1300$ and $w_i^{\text{sm-p}} = 100 \cdot |\mathcal{V}_i^{\text{fn}}|$ as objective function coefficients.

Third, for the IVAP, we used $\overline{w}^{\text{sm-p}} = 10^4$ and $\overline{w}^{\text{sm-q}} = 10^3$ as well as $w_v^{\text{sm-p}} = 10$ and $w_v^{\text{sm-q}} = 1$ for all $v \in \mathcal{V}^{\text{pi}}$. Additionally, we set $\varepsilon := 10^{-2} \frac{m}{s}$, $\gamma = 10$, $\mu = 3$, and $\Delta = 100$. Further, we decreased the threshold v^{min} to $= 10^{-3} \frac{m}{s}$.

Finally, we used $t_s = 8$ in equation (3.61) to scale the initial supplies and demands w.r.t. the four hydrogen test sets `H2`, `H2-B`, `H2-P`, and `H2-BP`. Thus, after linearly scaling up the boundary values for the first twelve hours, for the last twelve hours the supplies and demands were equal in terms of energy w.r.t. the original natural gas input data. However, we note that the supplies and demands were additionally balanced in test sets `H2-B` and `H2-BP`.

A.2. Results for NG-TC

The following tables contain the detailed results of the computational experiments for test set NG-TC, which are described and discussed in Section 3.10. The first column contains the instance name. It consists of the virtual day and the time in hours and minutes of the corresponding initial state, i.e., 2-0400 is the instance with the network state from 4AM on virtual day 2. The second column states the gap of the MIP solve in % that was performed last. The third column states the runtime of the complete Algorithm 4. While the fourth column denotes the total number of conducted simple state changes in the final solution, the fifth column counts only those for which a flow direction change was simultaneously performed in the same network station and in the same time step. The sixth column states the amount of used compression energy in MWh, which we calculated a posteriori using the exact nonlinear power equation for turbo compressor units (P). Finally, while the seventh column denotes the flow imbalance over the considered time horizon in kg, the last two columns state the additional number of iterations of Algorithm 4 that had to be performed as well as the number of added no-good cuts, respectively.

Instance	Gap	Time	S	S+F	Energy	Imbalance	Iter	NG
1-1200	0	153	0	0	491	937 587	0	0
1-1230	0	177	0	0	439	957 877	0	0
1-1300	0	156	0	0	468	1 150 775	0	0
1-1330	0	144	0	0	479	1 372 272	0	0
1-1400	0	195	0	0	506	1 450 230	0	0
1-1430	0	170	0	0	498	1 469 730	0	0
1-1500	0	198	0	0	433	1 637 187	0	0
1-1530	0	166	0	0	442	1 665 136	0	0
1-1600	0	162	0	0	442	1 787 531	0	0
1-1630	0	146	0	0	435	1 838 633	0	0
1-1700	0	133	0	0	430	1 828 104	0	0
1-1730	0	179	0	0	445	1 824 838	0	0
1-1800	0	181	0	0	453	1 757 738	0	0
1-1830	0	172	0	0	433	1 725 303	0	0
1-1900	0	226	0	0	453	1 640 375	0	0
1-1930	0	176	0	0	441	1 605 609	0	0
1-2000	0	171	0	0	441	1 490 590	0	0
1-2030	0	126	0	0	434	1 430 600	0	0
1-2100	0	224	0	0	444	1 314 478	0	0
1-2130	0	173	0	0	446	1 256 009	0	0
1-2200	0	252	0	0	451	1 113 943	0	0
1-2230	0	153	0	0	418	1 071 276	0	0
1-2300	0	165	0	0	433	963 572	0	0
1-2330	0	165	0	0	434	925 870	0	0

Instance	Gap	Time	S	S+F	Energy	Imbalance	Iter	NG
2-0000	0	200	0	0	436	865 785	0	0
2-0030	0	150	0	0	465	825 357	0	0
2-0100	0	141	0	0	407	747 585	0	0
2-0130	0	161	0	0	406	712 351	0	0
2-0200	0	149	0	0	413	647 573	0	0
2-0230	0	163	0	0	413	624 758	0	0
2-0300	0	127	0	0	397	529 154	0	0
2-0330	0	151	0	0	409	477 770	0	0
2-0400	0	225	0	0	406	370 497	0	0
2-0430	0	148	0	0	409	310 279	0	0
2-0500	0	131	0	0	396	156 607	0	0
2-0530	0	142	0	0	388	127 120	0	0
2-0600	0	169	0	0	392	84 815	0	0
2-0630	0	226	0	0	406	96 127	0	0
2-0700	0	175	0	0	393	198 562	0	0
2-0730	0	156	0	0	403	247 899	0	0
2-0800	0	168	0	0	415	353 511	0	0
2-0830	0	134	0	0	370	422 181	0	0
2-0900	0	170	0	0	364	462 583	0	0
2-0930	0	135	0	0	367	496 565	0	0
2-1000	0	131	0	0	358	664 575	0	0
2-1030	0	119	0	0	378	664 042	0	0
2-1100	0	121	0	0	380	644 885	0	0
2-1130	0	132	0	0	378	624 691	0	0
2-1200	0	127	0	0	351	651 113	0	0
2-1230	0	128	0	0	344	647 577	0	0
2-1300	0	143	0	0	344	700 762	0	0
2-1330	0	167	0	0	339	672 938	0	0
2-1400	0	142	0	0	340	704 869	0	0
2-1430	0	203	0	0	331	703 785	0	0
2-1500	0	203	0	0	315	735 565	0	0
2-1530	0	166	0	0	322	709 579	0	0
2-1600	0	160	0	0	316	680 715	0	0
2-1630	0	165	0	0	326	700 938	0	0
2-1700	0	170	0	0	327	738 743	0	0
2-1730	0	176	0	0	315	728 191	0	0
2-1800	0	157	0	0	310	464 387	0	0

Instance	Gap	Time	S	S+F	Energy	Imbalance	Iter	NG
2-1830	0	173	0	0	311	360 858	0	0
2-1900	0	177	0	0	299	235 634	0	0
2-1930	0	164	0	0	295	194 344	0	0
2-2000	19	21 796	1	1	295	141 549	0	0
2-2030	0	6322	1	1	292	117 106	0	0
2-2100	0	1712	1	1	288	26 774	0	0
2-2130	4	21 770	1	1	289	−15 695	0	0
2-2200	0	11 274	1	1	272	−108 674	0	0
2-2230	4	21 774	1	1	290	−152 307	0	0
2-2300	99	21 767	1	1	283	−291 601	0	0
2-2330	4	21 770	1	1	276	−328 056	0	0
3-0000	0	217	0	0	290	−409 889	0	0
3-0030	0	4645	1	1	292	−419 117	0	0
3-0100	0	208	0	0	281	−461 657	0	0
3-0130	0	2520	1	1	256	−478 710	0	0
3-0200	0	143	0	0	269	−520 195	0	0
3-0230	4	21 776	1	1	328	−534 862	0	0
3-0300	0	171	0	0	261	−533 922	0	0
3-0330	0	193	0	0	260	−534 444	0	0
3-0400	0	144	0	0	282	−536 085	0	0
3-0430	0	3792	1	1	259	−545 080	0	0
3-0500	0	157	0	0	258	−568 726	0	0
3-0530	0	154	0	0	243	−578 520	0	0
3-0600	0	199	0	0	280	−620 311	0	0
3-0630	0	181	0	0	255	−643 576	0	0
3-0700	0	162	0	0	254	−786 552	0	0
3-0730	0	164	0	0	263	−880 228	0	0
3-0800	0	175	0	0	270	−1 026 394	0	0
3-0830	0	176	0	0	270	−1 026 394	0	0
3-0900	0	174	0	0	261	−1 205 227	0	0
3-0930	0	141	0	0	269	−1 233 964	0	0
3-1000	0	145	0	0	257	−1 051 462	0	0
3-1030	0	179	0	0	267	−917 114	0	0
3-1100	0	168	0	0	284	−803 867	0	0
3-1130	0	168	0	0	227	−711 773	0	0
3-1200	0	149	0	0	215	−536 546	0	0

Instance	Gap	Time	S	S+F	Energy	Imbalance	Iter	NG
3-1230	0	172	0	0	223	−397 205	0	0
3-1300	0	145	0	0	222	−276 225	0	0
3-1330	0	149	0	0	222	−200 417	0	0
3-1400	0	169	0	0	217	−163 930	0	0
3-1430	0	145	0	0	214	−104 913	0	0
3-1500	0	165	0	0	218	−15 604	0	0
3-1530	0	155	0	0	230	29 308	0	0
3-1600	0	154	0	0	220	69 446	0	0
3-1630	0	188	0	0	225	73 285	0	0
3-1700	0	142	0	0	226	100 714	0	0
3-1730	0	171	0	0	221	145 056	0	0
3-1800	0	155	0	0	217	409 485	0	0
3-1830	0	142	0	0	206	549 527	0	0
3-1900	0	150	0	0	224	672 859	0	0
3-1930	0	153	0	0	212	739 446	0	0
3-2000	0	137	0	0	216	816 256	0	0
3-2030	0	146	0	0	276	892 373	0	0
3-2100	0	163	1	0	279	1 024 963	0	0
3-2130	0	165	1	1	267	1 078 579	0	0
3-2200	0	167	1	1	263	1 204 527	0	0
3-2230	0	160	1	0	262	1 286 348	0	0
3-2300	0	136	0	0	272	1 461 524	0	0
3-2330	0	148	0	0	246	1 511 868	0	0
4-0000	0	159	0	0	248	1 622 832	0	0
4-0030	0	124	0	0	248	1 606 114	0	0
4-0100	0	135	0	0	251	1 590 899	0	0
4-0130	0	122	0	0	254	1 561 612	0	0
4-0200	0	168	0	0	216	1 527 549	0	0
4-0230	0	146	0	0	208	1 474 785	0	0
4-0300	0	146	0	0	260	1 429 830	0	0
4-0330	0	127	0	0	222	1 402 420	0	0
4-0400	0	136	0	0	220	1 395 894	0	0
4-0430	0	141	0	0	229	1 396 321	0	0
4-0500	0	140	0	0	218	1 389 218	0	0
4-0530	0	180	0	0	227	1 326 282	0	0
4-0600	0	116	0	0	238	1 254 206	0	0

Instance	Gap	Time	S	S+F	Energy	Imbalance	Iter	NG
4-0630	0	147	0	0	225	1 267 305	0	0
4-0700	0	140	0	0	233	1 448 212	0	0
4-0730	0	150	0	0	224	1 504 836	0	0
4-0800	0	171	0	0	222	1 561 978	1	0
4-0830	0	136	0	0	219	1 499 749	0	0
4-0900	0	186	0	0	218	1 304 440	1	0
4-0930	0	128	0	0	262	1 085 166	0	0
4-1000	0	192	0	0	263	588 151	0	0
4-1030	0	182	0	0	259	306 108	0	0
4-1100	0	158	0	0	268	−42 758	0	0
4-1130	0	179	0	0	264	−283 713	0	0
4-1200	0	194	0	0	264	−686 161	0	0
4-1230	0	178	0	0	299	−991 019	0	0
4-1300	0	200	0	0	278	−1 361 054	0	0
4-1330	0	173	0	0	291	−1 574 419	0	0
4-1400	0	174	0	0	310	−1 792 112	0	0
4-1430	0	191	0	0	302	−1 941 797	0	0
4-1500	0	229	0	0	310	−2 156 480	0	0
4-1530	0	180	0	0	306	−2 224 473	0	0
4-1600	0	157	0	0	332	−2 327 456	0	0
4-1630	0	171	0	0	321	−2 387 498	0	0
4-1700	0	196	0	0	348	−2 548 052	0	0
4-1730	0	189	0	0	345	−2 694 243	0	0
4-1800	0	188	0	0	369	−2 824 433	0	0
4-1830	0	187	0	0	372	−2 893 266	0	0
4-1900	0	218	0	0	377	−2 973 482	0	0
4-1930	0	200	0	0	390	−3 008 679	0	0
4-2000	0	203	0	0	401	−3 027 965	0	0
4-2030	0	206	0	0	390	−3 046 369	0	0
4-2100	0	210	0	0	416	−3 048 737	0	0
4-2130	0	240	0	0	358	−3 015 735	0	0
4-2200	0	189	0	0	433	−3 020 750	0	0
4-2230	0	247	0	0	430	−2 992 784	0	0
4-2300	0	228	0	0	437	−2 959 635	0	0
4-2330	0	197	0	0	479	−2 957 380	0	0

Instance	Gap	Time	S	S+F	Energy	Imbalance	Iter	NG
6-0000	0	6943	1	0	394	−412 131	0	0
6-0030	0	214	0	0	397	−380 339	0	0
6-0100	0	317	0	0	407	−380 498	0	0
6-0130	0	5313	1	0	423	−293 519	0	0
6-0200	0	3560	1	1	387	−293 744	0	0
6-0230	0	557	0	0	394	−13 376	0	0
6-0300	0	212	0	0	409	−13 376	0	0
6-0330	0	1676	1	0	414	191 599	0	0
6-0400	0	11 353	1	0	418	191 127	0	0
6-0430	0	543	0	0	394	148 350	0	0
6-0500	0	284	0	0	381	148 495	0	0
6-0530	0	9988	1	0	404	189 638	1	0
6-0600	0	14 966	1	1	374	189 611	0	0
6-0630	0	200	0	0	355	152 315	0	0
6-0700	0	264	0	0	379	218 470	0	0
6-0730	0	984	1	0	368	396 488	0	0
6-0800	0	810	1	0	386	397 666	0	0
6-0830	0	211	0	0	371	536 175	0	0
6-0900	0	273	0	0	410	535 759	0	0
6-0930	0	783	1	1	403	734 803	0	0
6-1000	0	990	1	0	389	737 781	0	0
6-1030	0	205	0	0	395	748 308	0	0
6-1100	0	185	0	0	389	748 169	0	0
6-1130	0	946	1	0	408	694 453	1	0
6-1200	100	43 737	1	0	569	623 878	2	0
6-1230	100	86 460	1	0	526	581 705	4	0
6-1300	100	86 834	1	1	499	531 697	4	0
6-1330	0	3006	0	0	466	467 111	4	0
6-1400	0	525	0	0	490	361 884	2	0
6-1430	0	5209	0	0	479	297 829	4	1
6-1500	0	1145	0	0	388	361 819	2	0
6-1530	0	342	0	0	389	523 938	0	0
6-1600	0	271	0	0	404	793 081	0	0
6-1630	0	288	0	0	403	964 266	0	0
6-1700	0	264	0	0	435	1 169 320	0	0
6-1730	0	332	0	0	423	1 290 086	0	0
6-1800	0	294	0	0	395	1 492 078	0	0

Instance	Gap	Time	S	S+F	Energy	Imbalance	Iter	NG
6-1830	0	879	1	0	400	1 568 530	0	0
6-1900	0	600	1	0	408	1 662 643	0	0
6-1930	0	735	1	0	403	1 657 723	0	0
6-2000	0	4960	2	0	415	1 639 946	2	0
6-2030	0	966	1	1	413	1 676 963	0	0
6-2100	0	781	1	1	402	1 732 251	0	0
6-2130	0	1078	1	1	406	1 798 113	1	0
6-2200	0	2083	1	0	415	1 906 519	0	0
6-2230	0	1726	1	1	423	1 944 444	1	0
6-2300	0	20 571	1	1	424	1 992 298	7	1
6-2330	0	204	0	0	409	1 992 582	0	0
7-0000	0	221	0	0	428	1 959 515	0	0
7-0030	0	223	0	0	443	1 903 422	0	0
7-0100	0	245	0	0	429	1 774 190	0	0
7-0130	0	227	0	0	442	1 696 005	0	0
7-0200	0	214	0	0	444	1 586 070	0	0
7-0230	0	257	0	0	442	1 526 365	0	0
7-0300	0	261	0	0	424	1 441 760	0	0
7-0330	0	209	0	0	454	1 400 022	0	0
7-0400	0	271	0	0	467	1 370 517	0	0
7-0430	0	218	0	0	445	1 426 812	0	0
7-0500	0	232	0	0	453	1 529 520	0	0
7-0530	0	217	0	0	441	1 547 471	0	0
7-0600	0	274	0	0	448	1 584 125	0	0
7-0630	0	241	0	0	429	1 556 584	0	0
7-0700	0	243	0	0	442	1 509 339	0	0
7-0730	0	214	0	0	461	1 503 959	0	0
7-0800	0	202	0	0	430	1 487 131	0	0
7-0830	0	220	0	0	447	1 473 261	0	0
7-0900	0	170	0	0	449	1 478 892	0	0
7-0930	0	223	0	0	461	1 439 741	1	0
7-1000	0	224	0	0	492	1 434 509	0	0
7-1030	0	204	0	0	451	1 536 512	0	0
7-1100	0	246	0	0	434	1 682 756	0	0
7-1130	0	186	0	0	446	1 753 300	0	0
7-1200	0	182	0	0	445	1 847 006	0	0

Instance	Gap	Time	S	S+F	Energy	Imbalance	Iter	NG
7-1230	0	239	0	0	445	1 854 046	0	0
7-1300	0	199	0	0	444	1 854 363	0	0
7-1330	0	239	0	0	438	1 862 676	1	0
7-1400	0	174	0	0	440	1 830 463	0	0
7-1430	0	170	0	0	440	1 833 944	0	0
7-1500	0	234	0	0	424	1 787 920	0	0
7-1530	0	246	0	0	432	1 754 936	0	0
7-1600	0	161	0	0	420	1 627 178	0	0
7-1630	0	162	0	0	418	1 557 312	0	0
7-1700	0	183	0	0	433	1 475 887	0	0
7-1730	0	204	0	0	434	1 455 084	0	0
7-1800	0	167	0	0	423	1 424 055	0	0
7-1830	0	163	0	0	421	1 377 437	0	0
7-1900	0	159	0	0	411	1 308 076	0	0
7-1930	0	190	0	0	406	1 272 123	0	0
7-2000	0	162	0	0	399	1 209 144	0	0
7-2030	0	178	0	0	394	1 187 809	0	0
7-2100	0	156	0	0	406	1 175 733	0	0
7-2130	0	165	0	0	405	1 174 486	0	0
7-2200	0	167	0	0	403	1 179 305	0	0
7-2230	0	166	0	0	388	1 193 021	0	0
7-2300	0	160	0	0	373	1 209 263	0	0
7-2330	0	217	0	0	370	1 220 138	0	0
8-0000	0	186	0	0	416	1 236 968	0	0
8-0030	0	198	0	0	392	1 273 708	0	0
8-0100	0	202	0	0	407	1 400 688	0	0
8-0130	0	196	0	0	378	1 462 887	0	0
8-0200	0	169	0	0	369	1 527 455	0	0
8-0230	0	176	0	0	383	1 574 493	0	0
8-0300	0	185	0	0	363	1 661 806	0	0
8-0330	0	173	0	0	358	1 682 276	0	0
8-0400	0	134	0	0	338	1 659 721	0	0
8-0430	0	128	0	0	355	1 637 052	0	0
8-0500	0	172	0	0	332	1 608 532	0	0
8-0530	0	109	0	0	391	1 603 445	0	0
8-0600	0	179	0	0	375	1 588 751	0	0

Instance	Gap	Time	S	S+F	Energy	Imbalance	Iter	NG
8-0630	0	175	0	0	331	1 582 110	0	0
8-0700	0	126	0	0	328	1 690 406	0	0
8-0730	0	120	0	0	326	1 737 283	0	0
8-0800	0	136	0	0	304	1 799 537	0	0
8-0830	0	138	0	0	301	1 833 787	0	0
8-0900	0	144	0	0	321	1 875 427	0	0
8-0930	0	163	0	0	291	1 934 056	0	0
8-1000	0	160	0	0	288	2 010 785	0	0
8-1030	0	166	0	0	285	2 055 087	0	0
8-1100	0	156	0	0	277	2 217 295	0	0
8-1130	0	151	0	0	264	2 426 878	0	0
8-1200	0	165	0	0	274	2 788 246	0	0
8-1230	0	144	0	0	259	3 071 818	0	0
8-1300	0	171	0	0	286	3 472 413	0	0
8-1330	0	151	0	0	248	3 660 691	0	0
8-1400	0	128	0	0	271	3 949 092	0	0
8-1430	0	133	0	0	267	4 076 454	0	0
8-1500	0	148	0	0	258	4 275 609	0	0
8-1530	0	173	0	0	243	4 386 862	0	0
8-1600	0	165	0	0	240	4 598 516	0	0
8-1630	0	192	0	0	252	4 721 583	0	0
8-1700	0	130	0	0	239	4 858 012	0	0
8-1730	0	200	0	0	233	4 888 837	0	0
8-1800	0	184	0	0	220	4 904 134	0	0
8-1830	0	148	0	0	216	4 901 191	0	0
8-1900	0	151	0	0	213	4 902 014	0	0
8-1930	0	168	0	0	252	4 875 685	0	0
8-2000	0	144	0	0	259	4 849 345	0	0
8-2030	0	152	0	0	247	4 791 535	0	0
8-2100	0	166	0	0	258	4 697 251	0	0
8-2130	0	161	0	0	234	4 643 512	0	0
8-2200	0	193	0	0	228	4 586 176	0	0
8-2230	0	125	0	0	237	4 499 442	0	0
8-2300	0	139	0	0	234	4 336 682	0	0
8-2330	0	198	0	0	224	4 239 311	0	0

Instance	Gap	Time	S	S+F	Energy	Imbalance	Iter	NG
9-0000	0	169	0	0	251	4 141 105	0	0
9-0030	0	167	0	0	242	4 118 445	0	0
9-0100	0	137	0	0	222	4 057 389	0	0
9-0130	0	145	0	0	231	4 040 457	0	0
9-0200	0	175	0	0	236	4 062 370	0	0
9-0230	0	146	0	0	226	4 030 539	0	0
9-0300	0	153	0	0	234	3 924 725	0	0
9-0330	0	161	0	0	237	3 889 521	0	0
9-0400	0	158	0	0	228	3 862 687	0	0
9-0430	0	162	0	0	214	3 842 896	0	0
9-0500	0	187	0	0	230	3 777 439	0	0
9-0530	0	179	0	0	226	3 745 948	0	0
9-0600	0	130	0	0	208	3 707 384	0	0
9-0630	0	146	0	0	214	3 673 858	0	0
9-0700	0	204	0	0	198	3 596 362	0	0
9-0730	0	152	0	0	189	3 525 181	0	0
9-0800	0	166	0	0	224	3 430 246	0	0
9-0830	0	162	0	0	186	3 369 537	0	0
9-0900	0	185	0	0	210	3 280 417	0	0
9-0930	0	168	0	0	170	3 193 766	0	0

A.3. Results for H2-TC

The following tables contain the detailed results of the computational experiments for test set H2-TC, which are described and discussed in Section 3.10. The first column contains the instance name. It consists of the virtual day and the time in hours and minutes of the corresponding initial state, i.e., 2-0400 is the instance with the network state from 4AM on virtual day 2. The second column states the gap of the MIP solve in % that was performed last. The third column states the runtime of the complete Algorithm 4. The fourth column states the boundary flow slack in kg applied in the final considered solution. While the fifth column denotes the total number of conducted simple state changes, the sixth column counts only those for which a flow direction change was simultaneously performed in the same network station and in the same time step. The last two columns state the amount of used compression energy in MWh, which we calculated a posteriori using the exact nonlinear power equation for turbo compressor units (P), and the flow imbalance of the instance in kg, respectively.

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
1-1200	6	24 096	3 918 940	12	8	77	108 659
1-1230	6	24 536	3 988 020	9	5	46	111 428
1-1300	5	24 272	4 098 440	9	4	40	135 055
1-1330	5	47 981	4 005 927	9	5	38	160 376
1-1400	5	24 014	4 196 671	11	7	38	169 046
1-1430	5	24 210	4 111 694	14	9	70	171 421
1-1500	4	24 081	3 647 493	13	7	74	191 044
1-1530	4	23 724	4 054 344	15	9	62	194 522
1-1600	4	23 903	4 245 942	12	7	45	208 731
1-1630	4	23 825	4 166 782	12	6	72	215 274
1-1700	4	23 661	4 286 151	12	8	44	212 662
1-1730	16	23 349	4 310 012	10	6	38	210 074
1-1800	4	23 923	4 314 773	11	7	24	202 376
1-1830	4	23 920	4 238 777	13	7	25	198 873
1-1900	7	23 275	4 309 037	8	6	24	189 331
1-1930	7	23 724	4 298 291	7	5	19	185 204
1-2000	5	23 698	4 310 447	14	6	14	172 105
1-2030	5	23 447	4 292 547	14	9	24	165 165
1-2100	9	23 676	4 398 082	11	8	16	151 947
1-2130	9	86 491	4 477 811	12	7	24	145 180
1-2200	7	23 372	4 314 996	10	7	56	128 844
1-2230	7	22 965	4 398 769	10	6	73	123 922
1-2300	6	23 364	4 373 875	8	6	49	111 405
1-2330	6	23 533	4 348 852	10	5	50	107 219

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
2-0000	8	23 505	4 244 613	10	4	53	100 277
2-0030	9	23 486	4 267 154	10	5	48	95 590
2-0100	7	23 332	4 345 322	8	5	32	86 713
2-0130	7	22 940	4 361 591	8	3	30	82 613
2-0200	11	46 485	4 249 571	10	5	42	75 110
2-0230	10	23 323	4 300 273	11	6	104	72 474
2-0300	9	23 298	4 329 312	7	4	39	61 382
2-0330	9	23 044	4 205 723	8	4	43	55 359
2-0400	13	23 199	4 211 908	9	3	49	42 953
2-0430	11	23 407	4 286 186	7	4	59	35 974
2-0500	10	23 289	4 293 421	9	6	76	18 148
2-0530	11	46 577	4 244 747	10	4	88	14 731
2-0600	17	23 097	3 795 534	12	7	116	9829
2-0630	13	22 857	3 868 021	10	5	96	11 145
2-0700	13	23 267	4 184 324	9	5	106	23 023
2-0730	14	23 314	4 169 467	10	5	80	28 744
2-0800	24	23 530	4 047 616	10	4	122	40 858
2-0830	18	23 352	4 069 475	8	4	109	48 902
2-0900	24	23 409	3 950 231	9	7	125	53 690
2-0930	19	70 049	4 028 716	9	5	73	58 241
2-1000	31	23 450	3 965 432	8	5	127	76 918
2-1030	29	23 551	3 947 902	9	5	118	76 876
2-1100	30	23 547	3 909 170	8	4	132	74 826
2-1130	30	47 044	3 892 503	10	7	131	73 034
2-1200	48	47 196	3 841 064	10	6	130	75 426
2-1230	48	47 233	3 822 518	10	7	136	75 442
2-1300	99	47 206	3 714 885	8	6	154	81 898
2-1330	99	46 841	3 675 017	9	7	173	78 961
2-1400	95	47 045	3 670 007	9	7	161	82 074
2-1430	94	47 357	3 564 253	9	7	163	82 156
2-1500	99	49 486	3 523 941	9	5	141	85 850
2-1530	99	47 431	3 393 024	9	7	168	82 877
2-1600	94	71 243	3 291 104	9	5	124	79 443
2-1630	99	70 761	3 213 823	9	6	126	81 775
2-1700	99	46 601	3 140 203	9	7	141	86 196
2-1730	99	70 943	2 983 479	9	7	158	85 050
2-1800	99	47 592	3 099 187	13	7	157	54 282

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
2-1830	99	48 739	2 976 854	12	5	133	42 205
2-1900	97	24 436	3 080 905	11	6	152	27 585
2-1930	97	23 835	3 042 801	10	5	123	22 721
2-2000	100	24 096	3 080 798	11	6	167	16 552
2-2030	92	24 444	3 009 209	11	7	171	13 716
2-2100	100	24 028	2 937 592	11	8	212	3134
2-2130	99	24 426	2 977 590	11	7	190	−1846
2-2200	100	24 741	2 980 281	11	7	154	−12 776
2-2230	100	24 213	2 884 710	11	7	172	−17 829
2-2300	82	23 968	2 832 975	12	4	181	−34 301
2-2330	99	24 073	2 929 658	13	4	155	−38 623
3-0000	18	23 667	3 048 813	15	9	193	−48 280
3-0030	8	23 940	3 205 801	13	8	187	−49 430
3-0100	7	23 651	3 299 317	13	8	184	−53 977
3-0130	6	70 824	2 902 799	15	7	66	−56 094
3-0200	6	47 517	3 101 896	16	11	109	−60 903
3-0230	6	23 718	3 239 685	12	5	120	−62 536
3-0300	4	23 742	3 306 979	11	5	115	−62 352
3-0330	4	23 968	3 303 395	11	7	74	−62 384
3-0400	5	23 967	3 266 966	11	7	80	−62 547
3-0430	5	23 772	3 343 802	13	8	76	−63 529
3-0500	4	23 672	3 185 559	12	5	65	−66 297
3-0530	5	23 933	3 185 535	14	8	62	−67 443
3-0600	5	23 767	2 833 438	15	13	86	−72 326
3-0630	7	23 839	2 702 297	18	11	89	−74 997
3-0700	4	23 820	2 962 463	13	5	84	−91 772
3-0730	0	22 435	773 251	9	2	93	−102 661
3-0800	0	86 509	1 096 571	12	4	89	−119 647
3-0830	0	86 510	1 096 571	12	4	89	−119 647
3-0900	0	22 425	926 630	10	4	178	−140 902
3-0930	0	22 466	862 336	11	4	157	−144 744
3-1000	0	86 552	931 939	13	6	151	−122 668
3-1030	0	23 098	988 718	12	4	161	−107 113
3-1100	0	22 614	801 634	11	3	141	−94 310
3-1130	1	22 485	305 123	9	5	153	−83 026
3-1200	17	23 187	576 806	12	6	155	−62 984

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
3-1230	0	23 452	659 987	10	5	141	−46 161
3-1300	0	22 844	600 109	11	4	193	−32 108
3-1330	0	22 569	209 429	10	5	176	−23 330
3-1400	0	22 953	469 214	13	2	202	−19 102
3-1430	0	22 244	527 468	11	5	225	−12 246
3-1500	0	22 356	611 296	10	3	217	−1817
3-1530	0	86 495	210 479	10	4	221	3407
3-1600	0	22 235	479 220	12	5	229	8056
3-1630	0	22 548	537 118	11	4	244	8509
3-1700	0	22 201	556 501	11	3	247	11 693
3-1730	0	22 062	213 221	12	6	190	16 959
3-1800	0	22 072	449 972	12	4	256	48 355
3-1830	0	22 111	534 683	11	7	228	64 283
3-1900	0	22 046	562 778	11	4	224	79 035
3-1930	0	22 051	210 536	10	2	191	86 872
3-2000	0	21 990	470 034	12	3	245	95 943
3-2030	0	22 206	529 069	11	5	243	104 983
3-2100	0	21 901	499 488	11	3	234	120 646
3-2130	0	21 980	175 247	10	6	186	127 026
3-2200	0	22 024	544 446	11	5	176	141 850
3-2230	0	21 761	609 619	9	3	164	152 188
3-2300	0	10 671	526 282	8	2	164	172 286
3-2330	0	456	208 818	10	3	151	178 028
4-0000	0	21 793	508 563	10	1	162	191 179
4-0030	0	14 150	417 821	7	1	169	189 287
4-0100	0	11 161	324 748	9	4	190	187 668
4-0130	0	662	135 491	9	3	146	184 249
4-0200	0	9397	465 330	8	3	124	180 193
4-0230	0	3081	465 381	6	2	127	174 014
4-0300	0	21 792	407 805	8	2	137	168 700
4-0330	0	480	149 084	8	1	136	165 410
4-0400	0	11 845	237 125	7	1	176	164 319
4-0430	0	21 825	269 756	7	2	145	164 428
4-0500	0	1702	348 392	7	3	137	163 655
4-0530	0	1021	147 129	8	3	136	156 890
4-0600	0	11 989	244 384	6	1	159	148 258

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
4-0630	0	1136	106 684	7	3	160	149 214
4-0700	0	1441	295 526	6	2	127	170 571
4-0730	0	737	114 836	6	1	168	177 632
4-0800	0	702	130 356	6	2	123	184 110
4-0830	0	694	127 840	6	1	140	176 705
4-0900	0	1658	125 063	6	1	143	153 710
4-0930	0	1761	59 935	6	2	160	128 466
4-1000	0	824	78 537	6	1	137	69 704
4-1030	0	17 660	0	5	1	159	36 310
4-1100	0	3648	0	5	2	146	−5012
4-1130	0	30 630	0	5	2	154	−33 243
4-1200	0	27 472	0	5	2	160	−80 390
4-1230	0	7105	0	5	1	181	−116 126
4-1300	23	23 695	0	5	2	157	−159 546
4-1330	0	18 076	0	5	2	174	−184 847
4-1400	46	27 853	0	5	1	178	−210 481
4-1430	44	24 718	0	5	1	177	−227 879
4-1500	46	30 275	0	5	2	175	−253 014
4-1530	28	22 743	0	5	2	180	−261 979
4-1600	46	43 422	0	5	1	190	−273 584
4-1630	0	78 986	1239	5	2	190	−280 975
4-1700	25	75 263	11 183	5	1	186	−299 741
4-1730	0	72 271	16 560	5	2	178	−316 893
4-1800	0	54 994	8419	5	2	189	−332 034
4-1830	49	86 516	9327	5	2	198	−339 961
4-1900	0	54 190	9365	5	2	187	−349 435
4-1930	29	86 467	16 744	5	2	183	−353 749
4-2000	35	73 491	38 363	5	1	189	−356 692
4-2030	23	84 482	48 477	5	1	198	−359 025
4-2100	16	86 430	6424	5	2	193	−358 917
4-2130	0	83 112	9286	5	2	187	−355 127
4-2200	0	68 665	0	5	2	202	−353 558
4-2230	37	86 582	0	5	2	168	−350 585
4-2300	37	66 022	0	5	2	166	−348 497
4-2330	24	86 437	0	5	2	156	−346 338

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
6-0000	0	22 659	239 542	12	8	201	−47 740
6-0030	0	66 649	190 325	8	4	144	−43 946
6-0100	0	22 162	198 837	8	5	149	−43 946
6-0130	0	22 752	252 711	12	7	203	−33 954
6-0200	0	22 760	228 549	12	7	191	−33 954
6-0230	0	22 289	193 416	8	4	144	−1546
6-0300	0	22 131	193 221	8	4	142	−1546
6-0330	0	22 240	218 514	12	7	269	22 146
6-0400	0	22 338	214 862	12	8	262	22 146
6-0430	1	21 906	191 070	8	4	146	17 257
6-0500	1	22 050	173 671	8	2	178	17 257
6-0530	0	86 489	126 947	12	8	219	22 025
6-0600	0	21 927	176 220	13	7	198	22 025
6-0630	0	22 051	317 547	10	2	134	17 713
6-0700	0	86 557	161 585	8	1	172	25 457
6-0730	0	8367	80 306	6	3	182	46 146
6-0800	0	4401	80 306	6	3	158	46 146
6-0830	0	359	71 089	6	2	157	62 259
6-0900	0	1008	88 110	6	2	179	62 259
6-0930	0	86 444	35 554	9	4	204	85 646
6-1000	0	86 436	35 554	9	5	252	85 646
6-1030	1	21 893	228 668	10	6	163	86 849
6-1100	1	86 531	228 799	10	5	149	86 849
6-1130	0	86 432	162 146	9	2	145	80 803
6-1200	1	86 424	84 407	11	6	190	72 775
6-1230	0	20 257	0	5	2	210	67 924
6-1300	0	15 125	0	5	1	277	61 893
6-1330	0	43 809	83 642	5	1	224	54 410
6-1400	0	64 175	6647	5	1	353	42 333
6-1430	0	26 557	0	5	1	253	34 753
6-1500	79	86 418	107 545	7	4	149	42 136
6-1530	1	86 526	58 637	6	2	199	61 309
6-1600	35	45 262	48 536	7	3	145	92 934
6-1630	32	66 993	30 440	7	4	146	112 419
6-1700	21	45 122	56 877	7	4	146	135 988
6-1730	2	21 753	182 008	7	3	162	149 777
6-1800	44	43 375	46 498	7	3	168	173 170

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
6-1830	84	31 034	210 818	8	5	180	181 559
6-1900	66	51 261	91 259	8	4	164	192 243
6-1930	0	1709	29 508	5	3	182	191 720
6-2000	9	21 738	0	7	4	200	189 759
6-2030	0	14 080	47 037	5	3	237	194 229
6-2100	27	22 012	0	6	4	192	200 822
6-2130	26	22 094	0	6	5	183	208 609
6-2200	27	22 070	0	6	3	189	221 339
6-2230	40	22 010	0	6	5	181	225 661
6-2300	42	47 608	15 606	6	3	177	231 221
6-2330	75	22 344	0	7	3	179	231 616
7-0000	44	22 337	0	4	2	184	228 524
7-0030	45	22 178	0	4	2	174	220 801
7-0100	60	44 048	0	5	3	190	206 007
7-0130	44	21 957	0	4	3	185	197 131
7-0200	44	22 087	0	4	2	205	184 488
7-0230	0	658	0	3	1	178	177 394
7-0300	44	22 055	0	4	3	194	167 316
7-0330	25	22 062	0	3	2	169	162 370
7-0400	40	22 061	0	4	2	179	159 063
7-0430	0	156	0	3	2	187	165 449
7-0500	40	22 072	0	4	1	178	177 422
7-0530	0	485	0	3	2	185	179 615
7-0600	0	162	0	3	2	206	183 753
7-0630	0	134	0	3	2	167	181 551
7-0700	18	22 055	0	4	2	205	176 015
7-0730	0	272	0	3	2	203	174 156
7-0800	0	176	0	3	2	169	172 492
7-0830	18	22 044	0	4	1	185	170 943
7-0900	0	8707	0	4	1	167	171 656
7-0930	0	25 545	0	4	1	186	167 284
7-1000	20	22 029	0	4	2	167	166 693
7-1030	20	22 063	0	4	2	179	179 497
7-1100	22	22 168	0	4	2	182	195 345
7-1130	22	22 194	0	4	1	165	203 253
7-1200	45	22 628	0	5	3	162	214 524

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
7-1230	44	22 650	0	5	1	164	215 325
7-1300	97	24 335	1 020 667	8	3	185	215 108
7-1330	44	22 066	0	5	2	155	217 380
7-1400	98	26 387	1 058 808	8	3	181	212 525
7-1430	98	24 849	1 175 965	8	3	189	212 613
7-1500	96	24 176	1 230 254	8	2	185	207 426
7-1530	97	24 919	1 255 182	10	3	157	203 663
7-1600	59	24 158	1 238 947	11	2	159	188 892
7-1630	98	25 470	1 277 345	8	2	172	180 676
7-1700	56	23 916	1 424 928	11	3	263	171 180
7-1730	16	23 889	1 861 670	8	4	155	168 865
7-1800	56	47 206	984 095	8	5	149	165 243
7-1830	98	26 242	1 565 130	8	5	150	160 003
7-1900	97	47 319	977 721	8	2	143	151 932
7-1930	98	23 873	1 474 521	10	3	160	147 728
7-2000	92	46 601	723 040	12	4	164	140 529
7-2030	97	24 826	1 130 538	10	3	160	138 114
7-2100	91	23 627	1 184 558	6	2	160	136 810
7-2130	90	24 422	1 175 302	8	2	169	136 668
7-2200	88	24 346	1 271 026	8	2	162	137 491
7-2230	90	23 666	917 842	7	3	160	138 798
7-2300	80	46 861	431 412	6	2	154	140 389
7-2330	91	23 996	946 427	10	2	254	141 463
8-0000	88	24 187	848 920	9	2	262	143 412
8-0030	89	24 177	879 271	7	2	166	147 706
8-0100	85	23 615	919 288	7	2	162	162 764
8-0130	86	25 649	883 993	5	2	168	170 023
8-0200	93	24 404	866 104	5	2	156	177 398
8-0230	93	23 913	817 251	5	2	165	182 791
8-0300	96	23 900	767 121	5	2	149	192 962
8-0330	92	24 842	834 360	5	2	154	195 336
8-0400	95	24 871	642 333	5	2	140	192 778
8-0430	89	24 085	438 114	5	2	197	190 254
8-0500	93	23 493	449 133	5	2	189	187 001
8-0530	91	23 839	394 198	6	3	166	186 046
8-0600	88	23 557	332 682	6	3	196	184 540

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
8-0630	91	25 288	255 322	9	3	219	184 195
8-0700	86	27 223	228 718	7	2	204	196 707
8-0730	90	34 505	203 542	8	2	211	202 093
8-0800	42	24 277	156 082	9	3	193	209 292
8-0830	53	26 076	97 043	7	3	127	213 198
8-0900	8	86 516	33 173	6	2	131	218 061
8-0930	15	50 067	40 679	5	3	136	224 723
8-1000	0	45 612	31 778	4	2	135	233 656
8-1030	34	86 676	22 015	6	2	165	238 750
8-1100	24	65 790	40 101	7	2	144	257 563
8-1130	37	25 133	84 961	7	2	142	282 241
8-1200	47	65 804	88 561	7	2	162	324 816
8-1230	0	66 050	124 012	9	3	197	358 860
8-1300	1	86 552	135 044	9	4	215	405 887
8-1330	1	66 757	156 268	6	2	229	426 934
8-1400	2	22 450	336 344	6	3	115	459 974
8-1430	1	86 515	302 180	6	3	203	475 915
8-1500	1	86 521	337 241	7	2	123	497 880
8-1530	1	86 530	364 359	7	3	146	511 877
8-1600	1	86 429	498 793	8	3	94	537 500
8-1630	1	22 956	564 484	6	4	89	551 752
8-1700	1	23 163	588 071	6	3	93	566 743
8-1730	1	22 884	623 612	6	4	95	569 700
8-1800	1	22 969	769 997	6	3	91	572 070
8-1830	1	23 080	755 558	6	4	84	572 059
8-1900	1	22 936	783 011	7	3	99	572 307
8-1930	1	23 068	810 851	7	2	85	569 785
8-2000	1	23 228	952 444	7	2	80	567 156
8-2030	1	23 306	967 186	7	3	80	561 602
8-2100	1	22 844	980 584	8	3	85	550 431
8-2130	1	68 861	1 019 481	7	4	89	543 224
8-2200	1	22 986	1 134 107	7	1	61	534 997
8-2230	1	68 719	1 100 537	8	3	70	524 445
8-2300	0	23 355	1 083 086	8	4	69	505 467
8-2330	1	86 464	1 087 750	8	5	114	494 317

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
9-0000	0	86 425	1 126 051	7	3	114	481 847
9-0030	1	86 425	1 158 726	9	4	62	479 754
9-0100	1	23 303	1 160 206	9	5	59	472 289
9-0130	0	86 431	1 242 531	7	4	64	471 986
9-0200	1	86 429	1 209 383	9	5	56	473 695
9-0230	1	46 166	1 277 999	9	3	56	468 894
9-0300	1	22 901	1 303 593	8	4	54	458 252
9-0330	0	23 188	1 314 998	9	4	60	453 971
9-0400	1	23 017	1 342 475	9	5	52	450 731
9-0430	1	23 467	1 383 246	9	5	57	446 778
9-0500	1	23 366	1 445 593	8	5	50	439 101
9-0530	1	22 622	1 498 384	7	5	47	436 740
9-0600	1	22 545	1 371 765	7	3	34	432 035
9-0630	1	22 767	1 345 654	7	4	32	427 060
9-0700	1	23 161	1 400 312	10	7	41	419 863
9-0730	1	23 040	1 458 821	7	3	29	411 934
9-0800	0	45 616	1 325 050	7	4	31	401 090
9-0830	0	45 707	1 309 378	9	6	29	394 350
9-0900	1	22 983	1 345 638	10	6	35	384 087
9-0930	1	46 355	1 378 671	10	5	36	373 408

A.4. Results for H2-HC-EQ

The following tables contain the detailed results of the computational experiments for test set H2-HC-EQ, which are described and discussed in Section 3.10. The first column contains the instance name. It consists of the virtual day, and the time in hours and minutes of the corresponding initial state, i.e., 2-0400 is the instance having the initial state from 4AM of virtual day 2. The second column states the gap of the MIP solve in % that was performed last. The third column states the runtime of the complete Algorithm 4. The fourth column states the boundary flow slack in kg applied in the final considered solution. While the fifth column denotes the total number of conducted simple state changes, the sixth column counts only those for which a flow direction change was simultaneously performed in the same network station and in the same time step. The last two columns state the amount of used compression energy in MWh, which we calculated a posteriori using the exact nonlinear power equation for turbo compressor units (P), and the flow imbalance of the instance in kg, respectively.

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
1-1200	12	43 568	0	3	1	1812	421 688
1-1230	15	65 118	0	2	2	1900	430 550
1-1300	10	65 053	0	3	1	1900	468 299
1-1330	13	21 860	0	3	1	1948	513 584
1-1400	12	65 507	0	3	2	1931	536 159
1-1430	44	86 609	0	4	2	1584	541 295
1-1500	49	58 780	0	3	1	1795	565 504
1-1530	44	35 050	0	4	2	1807	571 352
1-1600	44	35 109	0	4	2	1728	584 697
1-1630	44	52 675	0	4	3	1730	584 303
1-1700	44	43 494	0	4	3	1675	557 165
1-1730	44	40 517	0	4	2	1677	543 744
1-1800	44	54 076	0	4	3	1919	515 907
1-1830	34	86 693	0	3	1	1885	504 971
1-1900	0	22 363	0	3	2	1861	473 779
1-1930	29	43 696	0	3	2	1868	458 004
1-2000	34	52 187	0	3	2	1824	411 780
1-2030	0	21 810	0	3	2	1796	387 060
1-2100	34	43 901	0	3	1	1802	338 842
1-2130	34	23 203	0	3	2	1792	320 216
1-2200	34	44 460	0	3	1	1806	275 409
1-2230	33	22 693	0	3	1	1780	260 648
1-2300	0	1994	0	2	0	1759	218 127
1-2330	0	1453	0	2	1	1861	205 624

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
2-0000	0	264	0	2	0	1797	191 852
2-0030	0	267	0	2	1	1790	182 084
2-0100	0	246	0	2	0	1752	154 060
2-0130	0	242	0	2	1	1774	143 990
2-0200	0	366	0	3	2	2064	134 096
2-0230	0	817	0	3	2	2026	132 436
2-0300	0	784	0	3	2	2266	105 463
2-0330	0	408	0	3	2	1650	91 437
2-0400	0	3232	0	3	1	1749	70 823
2-0430	0	699	0	3	2	1876	60 218
2-0500	0	3213	0	3	2	2065	25 495
2-0530	0	1975	0	3	2	1775	16 664
2-0600	0	1269	0	3	2	1866	14 458
2-0630	0	2101	0	3	2	1813	16 150
2-0700	0	1063	0	3	2	1753	45 926
2-0730	0	4350	0	3	2	1618	61 188
2-0800	0	630	0	3	1	1673	88 950
2-0830	0	644	0	3	2	1880	108 047
2-0900	0	419	0	3	1	1591	135 930
2-0930	0	800	0	3	1	1704	147 940
2-1000	0	743	0	3	2	1732	177 891
2-1030	0	457	0	3	2	1696	178 880
2-1100	0	451	0	3	2	1848	198 235
2-1130	0	501	0	3	2	1622	198 987
2-1200	0	1530	0	3	2	1632	207 889
2-1230	0	493	0	3	1	1762	207 995
2-1300	0	339	0	3	1	1671	237 899
2-1330	0	372	0	3	2	1628	240 882
2-1400	0	3144	0	3	2	1656	269 083
2-1430	0	326	0	3	2	1828	275 109
2-1500	0	5319	0	3	2	1580	292 683
2-1530	0	385	0	3	1	1612	289 323
2-1600	0	4640	0	3	1	1569	289 876
2-1630	0	4028	0	3	1	1604	295 941
2-1700	0	287	0	3	1	1479	305 198
2-1730	0	2416	0	3	2	1413	300 301
2-1800	0	523	0	3	2	1360	206 229

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
2-1830	0	1651	0	3	1	1507	171 279
2-1900	0	1384	0	3	1	1409	130 091
2-1930	0	287	0	3	2	1348	114 038
2-2000	0	1232	0	3	2	1260	84 244
2-2030	0	1227	0	3	1	1263	66 394
2-2100	0	311	0	3	1	1385	19 775
2-2130	0	663	0	3	1	1230	−464
2-2200	0	1101	0	3	1	1130	−48 388
2-2230	0	758	0	3	1	1215	−70 581
2-2300	0	945	0	3	2	1369	−138 978
2-2330	0	535	0	3	1	1287	−157 131
3-0000	0	547	0	3	2	1236	−191 191
3-0030	0	754	0	3	1	1216	−197 923
3-0100	0	491	0	3	2	1246	−221 151
3-0130	0	1014	0	3	2	1232	−231 406
3-0200	0	1229	0	3	1	1238	−248 023
3-0230	0	493	0	3	2	1113	−251 053
3-0300	0	509	0	3	2	1291	−259 381
3-0330	0	673	0	3	2	1248	−264 174
3-0400	0	752	0	3	1	1282	−273 247
3-0430	0	356	0	3	2	1410	−280 462
3-0500	0	2040	0	3	2	1450	−305 829
3-0530	0	617	0	3	1	1329	−315 787
3-0600	0	601	0	3	2	1434	−339 683
3-0630	0	367	0	3	2	1365	−348 107
3-0700	0	422	0	3	2	1380	−400 159
3-0730	0	1432	0	3	2	1440	−427 580
3-0800	0	581	0	3	2	1598	−467 193
3-0830	0	582	0	3	2	1598	−467 193
3-0900	0	521	0	2	2	1342	−506 959
3-0930	0	575	0	2	2	1581	−505 470
3-1000	0	800	0	2	2	1621	−422 805
3-1030	0	824	0	2	1	1511	−366 117
3-1100	0	422	0	2	2	1709	−304 120
3-1130	0	720	0	2	2	1141	−258 292
3-1200	0	470	0	2	2	1161	−179 362

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
3-1230	0	664	0	2	2	1161	−118 777
3-1300	0	522	0	2	2	1258	−45 652
3-1330	0	729	0	2	1	1178	−1530
3-1400	0	1690	0	2	2	1099	48 976
3-1430	0	525	0	2	1	1263	86 928
3-1500	0	232	0	1	1	1257	142 136
3-1530	0	290	0	1	1	1248	165 065
3-1600	0	226	0	1	1	1145	200 802
3-1630	0	340	0	1	1	1269	216 797
3-1700	0	240	0	1	0	1071	249 341
3-1730	0	271	0	1	1	1068	272 188
3-1800	0	629	0	1	1	1138	329 495
3-1830	0	266	0	1	1	1113	362 301
3-1900	0	247	0	1	1	1037	401 509
3-1930	0	281	0	1	1	968	419 486
3-2000	0	306	0	1	1	1064	440 155
3-2030	44	66 346	0	2	2	1338	456 655
3-2100	0	1472	0	2	1	1255	491 127
3-2130	0	10 627	0	2	2	1200	502 371
3-2200	0	18 831	0	2	1	1282	533 371
3-2230	0	26 484	0	2	1	1282	551 802
3-2300	0	32 212	0	2	1	1125	586 075
3-2330	0	26 274	0	2	1	1159	595 024
4-0000	0	59 572	0	2	0	1109	619 899
4-0030	2	48 843	0	2	1	1114	607 913
4-0100	39	21 814	0	2	1	1123	593 827
4-0130	0	13 456	0	2	1	1146	576 335
4-0200	0	28 413	0	2	1	1037	548 328
4-0230	0	5299	0	2	2	1129	522 243
4-0300	0	4744	0	2	2	1035	498 447
4-0330	0	16 916	0	2	0	1100	482 321
4-0400	0	11 434	0	2	1	1088	463 541
4-0430	0	44 064	0	2	2	964	448 161
4-0500	0	6711	0	2	1	991	424 710
4-0530	0	300	0	1	0	974	391 595
4-0600	0	240	0	1	0	985	342 883

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
4-0630	0	185	0	1	1	948	332 101
4-0700	0	395	0	1	1	937	368 170
4-0730	0	202	0	1	1	930	367 137
4-0800	0	357	0	1	1	998	353 137
4-0830	0	268	0	1	1	941	311 370
4-0900	0	296	0	1	1	891	206 669
4-0930	0	595	0	1	1	994	117 741
4-1000	0	333	0	1	1	1158	−37 321
4-1030	0	225	0	1	1	1209	−123 817
4-1100	0	258	0	1	1	1287	−231 631
4-1130	0	227	0	1	1	1359	−304 191
4-1200	0	323	0	1	1	1374	−422 774
4-1230	0	265	0	1	1	1466	−511 622
4-1300	43	22 692	0	2	1	1311	−613 531
4-1330	43	22 331	0	2	1	1451	−671 400
4-1400	0	427	0	1	0	1243	−732 058
4-1430	0	45 145	0	2	1	1404	−769 397
4-1500	11	38 605	0	2	2	1668	−823 725
4-1530	48	66 405	0	2	2	1822	−842 444
4-1600	11	86 449	1636	1	0	1712	−869 657
4-1630	0	27 796	47 925	1	1	1532	−887 273
4-1700	67	81 962	0	2	1	2360	−940 434
4-1730	88	86 620	0	7	5	2111	−987 021
4-1800	73	44 414	0	3	1	2304	−1 032 249
4-1830	73	44 425	0	3	1	2392	−1 050 887
4-1900	74	22 426	0	3	1	2480	−1 071 872
4-1930	75	44 604	0	3	2	2333	−1 079 959
4-2000	73	44 534	0	3	2	2724	−1 089 419
4-2030	74	44 291	0	3	1	2518	−1 097 059
4-2100	75	44 245	0	3	2	3115	−1 095 365
4-2130	60	66 876	0	3	1	3445	−1 083 522
4-2200	18	66 225	0	3	1	3123	−1 074 838
4-2230	61	44 330	0	3	1	3155	−1 060 043
4-2300	74	43 774	0	3	1	3217	−1 044 514
4-2330	72	44 028	0	3	2	3398	−1 026 087

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
6-0000	0	1199	0	3	1	2063	−168 101
6-0030	0	377	0	3	0	2272	−125 703
6-0100	0	563	0	3	1	2073	−125 703
6-0130	0	3496	0	3	1	2125	−85 628
6-0200	0	1422	0	3	2	1994	−85 628
6-0230	0	580	0	3	1	2223	−893
6-0300	0	578	0	3	1	2109	−893
6-0330	0	1130	0	3	2	2141	64 965
6-0400	0	657	0	3	2	2168	64 965
6-0430	0	3086	0	3	1	1945	70 211
6-0500	0	871	0	3	1	2264	70 211
6-0530	0	1852	0	3	1	2110	106 844
6-0600	0	583	0	3	2	2126	106 844
6-0630	0	520	0	3	1	2035	123 335
6-0700	0	386	0	3	1	2019	147 573
6-0730	0	466	0	3	1	2237	215 106
6-0800	0	1909	0	3	2	1976	215 106
6-0830	61	68 628	0	4	1	1894	269 142
6-0900	68	47 190	0	3	0	2154	269 142
6-0930	61	44 133	0	4	2	1948	348 771
6-1000	61	52 226	0	4	3	1888	348 771
6-1030	59	45 197	0	4	2	1948	340 505
6-1100	59	27 082	0	4	1	1613	340 505
6-1130	61	86 485	0	4	1	1959	325 773
6-1200	5	86 448	0	3	1	1920	309 076
6-1230	6	86 441	0	3	1	1827	297 295
6-1300	6	86 431	0	3	2	1745	281 896
6-1330	6	86 435	0	3	2	1827	270 218
6-1400	6	86 449	0	3	1	2079	260 492
6-1430	43	86 440	0	3	1	1843	255 025
6-1500	41	86 546	0	3	1	1824	269 484
6-1530	0	2284	0	1	1	1729	294 761
6-1600	46	86 448	0	3	2	1778	337 288
6-1630	46	43 643	0	3	0	1938	358 072
6-1700	41	65 939	811	3	1	1801	381 247
6-1730	42	65 666	645	3	1	1808	394 147
6-1800	42	65 601	652	3	1	2064	422 147

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
6-1830	31	44 174	615	4	1	1903	438 785
6-1900	46	34 463	650	4	1	1866	462 488
6-1930	33	86 632	641	4	2	1786	470 473
6-2000	52	67 087	565	5	2	1919	482 749
6-2030	54	86 672	24 313	4	3	1807	495 675
6-2100	42	86 626	26 703	4	1	1874	515 946
6-2130	32	86 444	589	4	1	1818	531 204
6-2200	55	86 578	590	4	2	1859	552 744
6-2230	56	65 341	598	4	2	1775	563 174
6-2300	59	86 607	25 286	4	1	1792	579 301
6-2330	55	86 666	625	4	1	1815	583 580
7-0000	46	65 890	646	3	1	1842	579 605
7-0030	46	86 605	658	3	1	1843	563 881
7-0100	46	86 568	676	4	1	2182	528 849
7-0130	28	86 688	25 370	3	1	1901	500 177
7-0200	24	86 667	40 848	3	1	1903	460 889
7-0230	4	22 048	723	3	2	1890	451 006
7-0300	4	22 019	740	3	0	1955	440 883
7-0330	6	43 775	751	3	1	1953	438 928
7-0400	4	21 962	768	3	1	1928	446 867
7-0430	4	21 963	761	3	1	1875	470 411
7-0500	4	43 822	751	3	1	1939	511 560
7-0530	6	86 731	736	3	2	1992	522 933
7-0600	6	21 962	511	3	2	1883	542 113
7-0630	4	45 220	513	3	1	1885	538 961
7-0700	52	86 466	5446	3	1	2033	532 849
7-0730	44	65 477	15 759	3	2	1943	533 304
7-0800	10	86 456	7664	3	1	1859	533 816
7-0830	43	86 650	742	4	3	1890	534 017
7-0900	56	86 626	708	4	3	1871	541 202
7-0930	56	70 708	694	4	2	1877	542 968
7-1000	57	86 446	673	4	2	1801	558 647
7-1030	57	86 655	692	4	2	1948	591 341
7-1100	57	66 152	687	4	2	1960	623 877
7-1130	58	86 651	697	4	3	1937	633 294
7-1200	58	86 469	664	4	2	1899	647 838

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
7-1230	58	86 477	660	4	2	1903	638 318
7-1300	28	66 526	660	4	2	1884	617 415
7-1330	55	86 625	746	4	2	1845	612 553
7-1400	54	46 368	1311	4	3	1807	582 059
7-1430	29	70 798	609	4	2	1865	572 114
7-1500	57	86 674	3736	4	3	1854	541 467
7-1530	27	46 123	661	4	2	1977	520 202
7-1600	33	86 696	6586	4	3	1907	462 470
7-1630	29	50 691	664	4	2	1814	427 210
7-1700	54	86 658	556	4	2	1919	383 423
7-1730	54	86 470	9269	3	2	2083	370 414
7-1800	0	13 768	25 222	3	2	1988	355 831
7-1830	0	3074	393	3	2	1985	345 176
7-1900	0	1293	332	3	2	1912	330 505
7-1930	0	1561	357	3	2	1942	325 703
7-2000	0	2887	6274	3	2	1860	316 659
7-2030	0	10 183	249	3	2	2089	315 644
7-2100	0	14 455	129	3	2	1951	323 383
7-2130	0	318	167	2	1	1786	330 707
7-2200	0	800	66	2	1	1803	341 571
7-2230	0	301	0	2	0	1817	353 935
7-2300	0	245	0	2	0	1716	374 228
7-2330	0	383	0	2	1	1641	389 140
8-0000	0	289	0	2	1	1734	404 080
8-0030	0	431	0	2	1	1651	411 925
8-0100	0	303	0	2	1	1647	443 340
8-0130	0	479	0	2	0	1626	451 453
8-0200	0	478	0	2	1	1550	451 809
8-0230	0	1945	0	2	0	1519	464 515
8-0300	0	1031	0	2	1	1523	493 553
8-0330	0	4607	0	2	1	1885	501 667
8-0400	36	25 046	0	3	1	1543	499 744
8-0430	0	1446	0	2	0	1466	503 261
8-0500	0	343	0	2	0	1429	515 749
8-0530	0	332	0	2	1	1707	521 780
8-0600	0	2086	0	2	1	1627	535 867

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
8-0630	0	19 949	0	3	1	1406	534 509
8-0700	0	3295	0	3	2	1522	574 149
8-0730	0	33 624	0	3	1	1528	594 440
8-0800	0	25 144	0	3	1	1444	623 264
8-0830	0	16 309	0	3	1	1459	638 968
8-0900	24	67 417	0	3	1	1509	662 791
8-0930	38	86 086	0	3	2	1285	687 248
8-1000	88	43 132	0	14	9	1096	732 672
8-1030	35	66 096	0	4	3	1109	766 761
8-1100	29	86 551	0	4	3	1323	841 468
8-1130	75	86 462	56 460	4	3	1333	920 051
8-1200	83	47 679	277 720	4	1	1459	1 062 235
8-1230	75	47 859	462 999	4	1	1551	1 161 941
8-1300	55	47 759	593 488	4	2	1420	1 296 826
8-1330	49	22 158	650 981	3	1	1461	1 357 700
8-1400	38	31 902	625 942	3	2	1540	1 445 475
8-1430	49	37 689	665 988	3	2	1596	1 489 579
8-1500	74	27 077	817 919	4	1	1849	1 539 617
8-1530	98	31 420	823 380	4	2	1340	1 569 065
8-1600	35	30 265	862 689	4	0	1621	1 613 890
8-1630	24	72 318	900 074	4	2	1272	1 633 645
8-1700	77	75 162	736 693	4	2	1030	1 644 957
8-1730	24	51 065	708 975	4	2	1070	1 643 884
8-1800	27	68 980	776 684	4	3	1089	1 643 781
8-1830	31	46 097	858 651	4	2	1076	1 639 362
8-1900	26	22 606	722 085	3	2	1087	1 631 411
8-1930	30	22 806	890 641	3	2	1073	1 622 010
8-2000	72	68 671	713 709	4	2	1056	1 612 670
8-2030	32	22 631	688 373	3	2	1001	1 593 477
8-2100	90	22 747	743 655	3	2	1009	1 558 237
8-2130	34	23 067	620 039	3	3	1065	1 538 186
8-2200	72	22 599	703 210	3	0	954	1 517 949
8-2230	69	22 721	671 400	5	4	1052	1 490 478
8-2300	37	24 812	704 412	3	2	870	1 438 243
8-2330	42	24 586	715 057	3	2	825	1 409 241

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
9-0000	38	24 691	569 070	3	2	827	1 372 108
9-0030	34	47 032	484 577	3	3	1025	1 360 573
9-0100	38	47 286	532 847	3	2	1033	1 335 136
9-0130	44	25 024	543 478	3	1	976	1 319 586
9-0200	55	46 558	401 928	3	1	973	1 295 948
9-0230	49	23 642	580 413	3	2	919	1 272 133
9-0300	44	46 413	475 909	3	1	899	1 232 558
9-0330	71	24 084	473 691	3	3	859	1 207 653
9-0400	64	23 839	424 109	3	1	824	1 172 457
9-0430	63	23 808	314 390	3	3	1099	1 147 827
9-0500	58	25 560	250 369	3	1	983	1 110 494
9-0530	61	86 456	173 805	4	2	818	1 087 135
9-0600	55	45 066	136 922	2	2	933	1 050 451
9-0630	51	67 088	105 912	3	1	838	1 019 131
9-0700	43	86 487	42 852	3	2	890	975 895
9-0730	26	86 448	22 265	4	1	809	942 052
9-0800	15	86 484	808	3	3	859	891 820
9-0830	29	65 165	804	3	3	748	858 214
9-0900	15	86 587	916	3	3	760	809 500
9-0930	14	47 072	1114	3	2	742	771 985

A.5. Results for H2-HC-EQ-B

The columns of the following tables contain the detailed results of the computational experiments for test set H2-HC-EQ-B described and discussed in Section 3.10. The first column contains the instance name. It consists of the virtual day, and the time in hours and minutes of the corresponding initial state, i.e., 2-0400 is the instance having the initial state from 4AM of virtual day 2. The second column states the gap of the MIP solve in % that was performed last. The third column states the runtime of the complete Algorithm 4. The fourth column states the boundary flow slack in kg applied in the final considered solution. While the fifth column denotes the total number of conducted simple state changes, the sixth column counts only those for which a flow direction change was simultaneously performed in the same network station and in the same time step. The last column states the amount of used compression energy in MWh, which we calculated a posteriori using the exact nonlinear power equation for turbo compressor units (P).

Instance	Gap	Time	Slack	S	S+F	Energy
1-1200	0	441	0	2	1	1761
1-1230	0	471	0	1	1	1625
1-1300	0	329	0	2	1	1596
1-1330	0	328	0	2	0	1743
1-1400	0	345	0	2	0	1674
1-1430	0	705	0	2	1	1591
1-1500	0	345	0	1	1	1575
1-1530	0	338	0	2	1	1658
1-1600	0	593	2243	2	1	1637
1-1630	0	1131	2126	2	1	1622
1-1700	0	576	1634	2	1	1551
1-1730	0	567	2081	2	0	1615
1-1800	0	1183	0	2	1	1721
1-1830	0	445	0	2	1	2282
1-1900	0	252	0	2	1	1963
1-1930	0	279	0	2	1	1877
1-2000	0	302	0	2	1	2028
1-2030	0	322	0	2	0	1885
1-2100	0	298	0	2	1	2214
1-2130	0	424	0	2	1	2032
1-2200	0	1184	0	2	1	1859
1-2230	0	463	0	2	1	2004
1-2300	0	223	0	2	1	1944
1-2330	0	208	0	2	1	1839

Instance	Gap	Time	Slack	S	S+F	Energy
2-0000	0	505	0	2	0	1836
2-0030	0	444	0	2	1	2052
2-0100	0	310	0	2	1	1803
2-0130	0	394	0	2	0	1749
2-0200	0	347	7295	2	0	1820
2-0230	0	715	7271	2	1	1940
2-0300	0	476	0	3	2	2560
2-0330	0	366	0	3	2	1825
2-0400	0	1746	0	3	2	2041
2-0430	0	2593	0	3	2	1719
2-0500	0	4674	0	3	2	1918
2-0530	0	2649	0	3	2	1895
2-0600	0	1981	0	3	2	1784
2-0630	0	1327	0	3	2	1675
2-0700	0	901	0	3	2	1822
2-0730	0	3329	0	3	2	1680
2-0800	0	416	0	3	2	1793
2-0830	0	714	0	3	2	1763
2-0900	0	1615	0	3	2	1942
2-0930	0	563	0	3	2	1847
2-1000	0	566	0	3	1	1831
2-1030	0	442	0	3	2	1794
2-1100	0	3621	0	3	1	1614
2-1130	0	1898	0	3	2	1877
2-1200	0	888	0	3	2	1745
2-1230	0	390	0	3	2	1885
2-1300	0	328	0	3	2	1696
2-1330	0	688	0	3	2	1787
2-1400	0	2713	0	3	2	1925
2-1430	0	359	0	3	2	1500
2-1500	0	1738	0	3	2	1577
2-1530	0	620	0	3	2	1676
2-1600	0	265	0	3	2	1548
2-1630	0	282	0	3	1	1701
2-1700	0	429	0	3	2	1547
2-1730	0	1938	0	3	2	1517
2-1800	0	1051	0	3	2	1330

Instance	Gap	Time	Slack	S	S+F	Energy
2-1830	0	336	0	3	2	1615
2-1900	0	297	0	3	1	1390
2-1930	0	3219	0	3	2	1362
2-2000	0	424	0	3	1	1465
2-2030	0	384	0	3	1	1293
2-2100	0	394	0	3	2	1309
2-2130	0	967	0	3	1	1301
2-2200	0	905	0	3	2	1135
2-2230	0	765	0	3	1	1326
2-2300	0	4175	0	3	2	1165
2-2330	0	6025	0	3	2	1233
3-0000	0	16 273	0	3	2	1105
3-0030	0	9636	0	3	1	1043
3-0100	0	4996	0	3	1	1071
3-0130	0	2316	0	3	2	1019
3-0200	0	735	0	3	2	1049
3-0230	0	2397	0	3	2	1055
3-0300	0	1217	0	3	1	1079
3-0330	0	1697	0	3	2	1067
3-0400	0	699	0	3	2	1052
3-0430	0	303	0	3	1	1007
3-0500	0	634	0	3	1	956
3-0530	0	1225	0	3	1	1033
3-0600	0	987	0	3	1	806
3-0630	0	1242	0	3	1	918
3-0700	0	462	0	3	2	1045
3-0730	0	850	0	3	2	942
3-0800	0	964	0	3	2	1056
3-0830	0	958	0	3	2	1056
3-0900	0	436	0	2	2	964
3-0930	0	652	0	2	2	1018
3-1000	0	251	0	2	1	1156
3-1030	0	627	0	2	1	1093
3-1100	0	385	0	2	1	1269
3-1130	0	2014	0	2	2	1059
3-1200	0	4809	0	2	2	957

Instance	Gap	Time	Slack	S	S+F	Energy
3-1230	0	434	0	2	2	1318
3-1300	0	437	0	2	2	1289
3-1330	0	287	0	2	2	1332
3-1400	0	1864	0	2	2	1176
3-1430	0	480	0	2	2	1175
3-1500	0	181	0	1	1	1419
3-1530	0	161	0	1	1	1236
3-1600	0	481	0	1	1	1114
3-1630	0	226	0	1	1	1244
3-1700	0	202	0	1	0	1237
3-1730	0	202	0	1	0	1245
3-1800	0	398	0	1	1	1007
3-1830	0	199	0	1	1	1176
3-1900	0	347	0	1	1	1228
3-1930	0	250	0	1	1	1133
3-2000	0	242	0	1	1	1149
3-2030	0	212	0	1	1	1637
3-2100	0	209	0	2	1	1593
3-2130	0	304	0	2	2	1347
3-2200	0	258	0	2	2	1165
3-2230	0	409	0	1	1	1430
3-2300	0	309	0	1	1	1477
3-2330	0	423	0	1	1	1210
4-0000	0	389	0	1	0	1253
4-0030	0	295	0	1	1	1222
4-0100	0	267	0	1	1	1424
4-0130	0	374	0	1	1	1180
4-0200	0	302	0	1	1	1198
4-0230	0	401	0	1	0	1087
4-0300	0	307	0	1	1	1235
4-0330	0	322	0	1	0	1116
4-0400	0	312	0	1	1	1087
4-0430	0	331	0	1	1	933
4-0500	0	366	0	1	1	985
4-0530	0	372	0	1	1	1025
4-0600	0	332	0	1	1	950

Instance	Gap	Time	Slack	S	S+F	Energy
4-0630	0	413	0	1	1	1004
4-0700	0	343	0	1	1	1017
4-0730	0	332	0	1	1	1080
4-0800	0	307	0	1	1	1074
4-0830	0	290	0	1	1	1078
4-0900	0	310	0	1	1	1182
4-0930	0	351	0	1	1	1188
4-1000	0	310	0	1	1	1191
4-1030	0	225	0	1	1	1205
4-1100	0	249	0	1	0	1207
4-1130	0	281	0	1	1	1100
4-1200	0	398	0	1	1	1195
4-1230	0	20 169	0	2	1	1237
4-1300	0	33 345	0	2	2	1169
4-1330	0	9681	0	2	1	1256
4-1400	0	12 425	0	2	0	1291
4-1430	0	6259	0	2	1	1416
4-1500	0	10 947	0	2	1	1280
4-1530	0	1522	0	2	0	1261
4-1600	0	13 811	0	2	2	1484
4-1630	0	3848	0	2	1	1452
4-1700	0	12 297	0	2	1	1491
4-1730	0	1640	26 653	1	1	1414
4-1800	0	7668	0	2	1	1691
4-1830	0	2099	0	2	0	1528
4-1900	0	3054	0	2	1	1560
4-1930	0	661	0	1	1	1541
4-2000	0	3759	0	2	2	1556
4-2030	0	2283	0	2	1	1701
4-2100	0	257	0	1	0	1769
4-2130	0	438	0	1	0	1749
4-2200	0	240	0	1	1	1387
4-2230	0	214	0	1	0	1610
4-2300	0	223	0	1	1	1751
4-2330	0	451	0	1	1	1686

Instance	Gap	Time	Slack	S	S+F	Energy
6-0000	0	1157	0	3	1	2069
6-0030	0	1314	0	3	1	2075
6-0100	0	350	0	3	1	2090
6-0130	0	1922	0	3	1	2167
6-0200	0	2269	0	3	2	2143
6-0230	0	630	0	3	2	2231
6-0300	0	465	0	3	1	2231
6-0330	0	882	0	3	1	2314
6-0400	0	563	0	3	1	2298
6-0430	0	306	0	3	0	2217
6-0500	0	328	0	3	2	2157
6-0530	0	1607	0	3	1	2408
6-0600	0	1159	0	3	2	2167
6-0630	0	675	0	3	1	2106
6-0700	0	347	0	3	1	2060
6-0730	0	471	0	3	1	2249
6-0800	0	422	0	3	2	2419
6-0830	0	254	0	3	1	2306
6-0900	0	347	0	3	1	2584
6-0930	0	979	0	3	1	2574
6-1000	0	442	0	3	2	2430
6-1030	0	442	0	3	1	2565
6-1100	0	423	0	3	1	2429
6-1130	0	908	0	3	1	2147
6-1200	0	9754	0	3	1	1976
6-1230	0	5512	0	3	1	1827
6-1300	0	24 891	0	3	1	2029
6-1330	0	30 157	0	3	1	1847
6-1400	42	22 712	0	3	0	1965
6-1430	42	86 824	0	3	0	1876
6-1500	0	73 395	0	3	1	1713
6-1530	0	1095	0	1	0	1983
6-1600	0	88 211	0	2	1	2019
6-1630	0	588	0	2	1	2125
6-1700	0	294	750	2	1	2071
6-1730	0	285	672	2	1	1869
6-1800	0	886	608	2	0	1941

Instance	Gap	Time	Slack	S	S+F	Energy
6-1830	0	929	626	3	1	1990
6-1900	0	863	670	3	1	1909
6-1930	0	936	561	3	2	1849
6-2000	0	1811	587	4	1	1933
6-2030	0	533	586	3	1	1940
6-2100	0	4217	3344	3	1	2037
6-2130	0	7746	5367	3	1	2120
6-2200	0	2701	7210	3	1	2032
6-2230	0	17 070	16 074	3	1	2402
6-2300	0	4551	15 808	3	1	2045
6-2330	0	1606	22 087	3	1	2079
7-0000	0	1885	15 697	2	1	2318
7-0030	0	3615	18 016	2	0	2099
7-0100	0	2010	25 508	2	1	2120
7-0130	0	1459	17 717	2	1	1995
7-0200	0	1579	29 715	2	1	2140
7-0230	0	4977	14 427	2	1	2289
7-0300	0	1039	28 111	2	0	2312
7-0330	0	956	19 842	2	0	2139
7-0400	0	2640	21 554	2	1	2089
7-0430	0	2908	25 342	2	1	2070
7-0500	0	1561	673	2	1	1989
7-0530	0	26 695	760	2	1	2068
7-0600	0	1460	528	2	1	2514
7-0630	0	1245	93 820	2	1	2134
7-0700	0	2634	773	2	1	2815
7-0730	0	1413	782	2	0	2135
7-0800	0	956	717	2	1	1940
7-0830	0	1461	700	2	1	1982
7-0900	0	932	740	2	1	1806
7-0930	0	770	711	2	0	1935
7-1000	0	2320	669	2	1	1833
7-1030	0	6000	714	2	1	1839
7-1100	0	913	711	2	0	1810
7-1130	0	3314	111 355	2	1	1958
7-1200	0	706	737	2	1	2005

Instance	Gap	Time	Slack	S	S+F	Energy
7-1230	0	856	748	2	1	2196
7-1300	0	1096	666	2	1	1754
7-1330	0	492	55 454	2	1	2063
7-1400	0	1325	23 038	2	1	1869
7-1430	0	915	21 617	2	1	1948
7-1500	0	2789	20 386	2	1	1823
7-1530	0	882	13 937	2	0	1848
7-1600	0	3764	7774	2	1	1818
7-1630	0	655	7808	2	1	2165
7-1700	0	273	590	2	1	2197
7-1730	0	305	600	2	1	1817
7-1800	0	449	615	2	1	1869
7-1830	0	405	379	2	0	1849
7-1900	0	331	369	2	1	1943
7-1930	0	649	333	2	1	1732
7-2000	0	2344	380	2	0	1748
7-2030	0	422	107	2	1	1912
7-2100	0	475	131	2	1	1631
7-2130	0	333	171	2	1	2031
7-2200	0	286	170	2	1	2169
7-2230	0	359	0	2	1	1599
7-2300	0	368	0	2	0	1824
7-2330	0	336	0	2	1	1778
8-0000	0	235	0	2	0	1756
8-0030	0	616	0	2	1	1467
8-0100	0	363	0	2	1	1587
8-0130	0	549	0	2	0	1859
8-0200	0	270	0	2	0	1679
8-0230	0	313	0	2	0	1543
8-0300	0	437	0	2	1	1839
8-0330	0	361	0	2	1	1915
8-0400	0	279	0	2	1	1590
8-0430	0	332	0	2	1	1862
8-0500	0	244	0	2	1	2306
8-0530	0	361	0	2	1	1480
8-0600	0	302	0	2	1	2009

Instance	Gap	Time	Slack	S	S+F	Energy
8-0630	0	244	0	2	1	1865
8-0700	0	218	0	2	1	1619
8-0730	0	240	0	2	1	2143
8-0800	0	262	0	2	1	1996
8-0830	0	308	0	2	0	1540
8-0900	0	383	0	2	0	1610
8-0930	0	301	0	2	1	1580
8-1000	0	247	0	2	1	1899
8-1030	0	224	0	2	1	1887
8-1100	0	225	0	2	1	1593
8-1130	0	251	0	2	1	2101
8-1200	0	244	0	2	0	2276
8-1230	0	347	0	2	0	1809
8-1300	0	247	0	2	1	1861
8-1330	0	221	0	2	0	2097
8-1400	0	266	0	2	1	1888
8-1430	0	369	0	2	1	1808
8-1500	0	314	0	2	1	1641
8-1530	0	332	0	2	1	1978
8-1600	0	395	0	2	1	1823
8-1630	0	260	0	2	1	1744
8-1700	0	255	0	2	1	1896
8-1730	0	282	0	2	1	1692
8-1800	0	364	0	2	1	1607
8-1830	0	275	0	2	1	1555
8-1900	0	325	2085	1	1	1597
8-1930	0	211	1356	1	1	1816
8-2000	0	225	1341	1	1	1644
8-2030	0	207	1342	1	0	1576
8-2100	0	207	1342	1	0	1508
8-2130	0	207	1344	1	1	1441
8-2200	0	221	1346	1	1	1437
8-2230	0	225	1347	1	1	1571
8-2300	0	205	1348	1	1	1475
8-2330	0	236	1348	1	0	1461

Instance	Gap	Time	Slack	S	S+F	Energy
9-0000	0	224	1348	1	1	1495
9-0030	0	205	1347	1	1	1501
9-0100	0	214	1347	1	1	1410
9-0130	0	230	1347	1	1	1376
9-0200	0	409	1347	1	0	1334
9-0230	0	244	1346	1	1	1413
9-0300	0	261	1344	1	1	1303
9-0330	0	353	1343	1	1	1266
9-0400	0	230	1341	1	1	1209
9-0430	0	243	1337	1	0	1281
9-0500	0	249	1331	1	1	1040
9-0530	0	221	1546	1	1	1085
9-0600	0	210	1273	1	0	1071
9-0630	0	206	1268	1	0	1200
9-0700	0	209	1261	1	1	1113
9-0730	0	197	1257	1	0	1120
9-0800	0	206	844	1	1	1138
9-0830	0	218	839	1	0	1129
9-0900	0	204	832	1	1	1052
9-0930	0	207	829	1	1	904

A.6. Results for H2-HC-EQ-P

The columns of the following tables contain the detailed results of the computational experiments for test set H2-HC-EQ-P described and discussed in Section 3.10. The first column contains the instance name. It consists of the virtual day, and the time in hours and minutes of the corresponding initial state, i.e., 2-0400 is the instance having the initial state from 4AM of virtual day 2. The second column states the gap of the MIP solve in % that was performed last. The third column states the runtime of the complete Algorithm 4. The fourth column states the boundary flow slack in kg applied in the final solution. While the fifth column denotes the total number of conducted simple state changes, the sixth column counts only those for which a flow direction change was simultaneously performed in the same network station and in the same time step. The last two columns state the amount of used compression energy in MWh, which we calculated a posteriori using the exact nonlinear power equation for turbo compressor units (P) and the flow imbalance of the instance in kg, respectively.

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
1-1200	14	22 108	0	3	2	1773	421 688
1-1230	17	22 605	0	2	2	1842	430 550
1-1300	14	23 498	0	3	1	1824	468 299
1-1330	14	21 908	0	3	2	1823	513 584
1-1400	12	43 541	0	3	2	1812	536 159
1-1430	12	22 365	0	3	2	1833	541 295
1-1500	17	25 284	0	2	1	1836	565 504
1-1530	0	31 830	0	3	2	1817	571 352
1-1600	13	43 435	0	3	2	1834	584 697
1-1630	14	22 061	0	3	2	1787	584 303
1-1700	0	43 740	0	3	2	1778	557 165
1-1730	14	21 937	0	3	2	1754	543 744
1-1800	0	64 338	0	3	2	1690	515 907
1-1830	36	23 438	0	3	2	1693	504 971
1-1900	0	273	0	2	1	1868	473 779
1-1930	0	820	0	2	1	1747	458 004
1-2000	0	402	0	2	0	1786	411 780
1-2030	0	336	0	2	1	1779	387 060
1-2100	0	313	0	2	1	1742	338 842
1-2130	0	310	0	2	0	1684	320 216
1-2200	0	247	0	2	0	1728	275 409
1-2230	0	485	0	2	1	1720	260 648
1-2300	0	257	0	2	1	1743	218 127
1-2330	0	273	0	2	1	1760	205 624

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
2-0000	0	308	0	2	1	1747	191 852
2-0030	0	234	0	2	1	1775	182 084
2-0100	0	236	0	2	1	1640	154 060
2-0130	0	230	0	2	1	1698	143 990
2-0200	0	229	0	2	1	1742	134 096
2-0230	0	246	0	2	1	1700	132 436
2-0300	0	214	0	2	0	1891	105 463
2-0330	0	219	0	2	1	1777	91 437
2-0400	0	270	0	2	1	1675	70 823
2-0430	0	242	0	2	0	1689	60 218
2-0500	0	217	0	2	0	1684	25 495
2-0530	0	224	0	2	0	1742	16 664
2-0600	0	234	0	2	1	1715	14 458
2-0630	0	215	0	2	1	1615	16 150
2-0700	0	398	0	3	2	1747	45 926
2-0730	0	1025	0	3	1	1698	61 188
2-0800	0	1234	0	3	2	1539	88 950
2-0830	0	995	0	3	2	1563	108 047
2-0900	0	1018	0	3	2	1673	135 930
2-0930	0	1199	0	3	2	1628	147 940
2-1000	0	633	0	3	2	1746	177 891
2-1030	0	930	0	3	1	1767	178 880
2-1100	0	761	0	3	2	1472	198 235
2-1130	0	957	0	3	2	1600	198 987
2-1200	0	751	0	3	2	1557	207 889
2-1230	0	821	0	3	2	1585	207 995
2-1300	0	1018	0	3	2	1498	237 899
2-1330	0	340	0	3	1	1482	240 882
2-1400	0	551	0	3	2	1394	269 083
2-1430	0	647	0	3	1	1688	275 109
2-1500	0	3281	0	3	1	1670	292 683
2-1530	0	502	0	3	2	1607	289 323
2-1600	0	681	0	3	1	1458	289 876
2-1630	0	310	0	3	2	1453	295 941
2-1700	0	1036	0	3	2	1445	305 198
2-1730	0	806	0	3	1	1298	300 301
2-1800	0	715	0	3	2	1415	206 229

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
2-1830	0	3480	0	3	2	1325	171 279
2-1900	0	393	0	3	2	1299	130 091
2-1930	0	1439	0	3	2	1244	114 038
2-2000	0	454	0	3	2	1109	84 244
2-2030	0	747	0	3	2	1394	66 394
2-2100	0	305	0	3	2	1196	19 775
2-2130	0	571	0	3	1	1310	−464
2-2200	0	415	0	3	1	1275	−48 388
2-2230	0	324	0	3	2	1252	−70 581
2-2300	0	389	0	3	2	1312	−138 978
2-2330	0	501	0	3	2	1211	−157 131
3-0000	0	329	0	3	2	1300	−191 191
3-0030	0	447	0	3	2	1315	−197 923
3-0100	0	380	0	3	2	1153	−221 151
3-0130	0	313	0	3	2	1107	−231 406
3-0200	0	417	0	3	1	1177	−248 023
3-0230	0	503	0	3	1	1156	−251 053
3-0300	0	406	0	3	2	1153	−259 381
3-0330	0	337	0	3	2	1119	−264 174
3-0400	0	363	0	3	2	1415	−273 247
3-0430	0	278	0	3	2	1180	−280 462
3-0500	0	295	0	3	2	1399	−305 829
3-0530	0	378	0	3	2	1430	−315 787
3-0600	0	413	0	3	1	1381	−339 683
3-0630	0	427	0	3	2	1340	−348 107
3-0700	0	309	0	3	2	1361	−400 159
3-0730	0	978	0	3	2	1527	−427 580
3-0800	0	322	0	3	1	1838	−467 193
3-0830	0	324	0	3	1	1838	−467 193
3-0900	0	302	0	2	2	1547	−506 959
3-0930	0	828	0	2	2	1642	−505 470
3-1000	0	13 927	0	2	2	1550	−422 805
3-1030	0	229	0	2	2	1529	−366 117
3-1100	0	180	0	1	1	1593	−304 120
3-1130	0	185	0	1	1	1284	−258 292
3-1200	0	175	0	1	1	1246	−179 362

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
3-1230	0	191	0	1	1	1488	−118 777
3-1300	0	226	0	1	1	1263	−45 652
3-1330	0	180	0	1	1	1339	−1530
3-1400	0	175	0	1	1	1220	48 976
3-1430	0	161	0	1	1	1131	86 928
3-1500	0	184	0	1	1	1337	142 136
3-1530	0	191	0	1	1	1339	165 065
3-1600	0	196	0	1	1	1155	200 802
3-1630	0	170	0	1	1	1157	216 797
3-1700	0	195	0	1	1	1210	249 341
3-1730	0	203	0	1	1	1158	272 188
3-1800	0	196	0	1	1	1084	329 495
3-1830	0	210	0	1	1	1030	362 301
3-1900	0	201	0	1	1	1072	401 509
3-1930	0	228	0	1	1	1006	419 486
3-2000	0	213	0	1	1	966	440 155
3-2030	0	216	0	1	1	1248	456 655
3-2100	0	258	0	1	0	1262	491 127
3-2130	0	256	0	1	1	1187	502 371
3-2200	0	246	0	1	1	1134	533 371
3-2230	0	237	0	1	1	1084	551 802
3-2300	0	298	0	1	0	1109	586 075
3-2330	0	277	0	1	0	1051	595 024
4-0000	0	370	0	1	1	1078	619 899
4-0030	0	280	0	1	1	1088	607 913
4-0100	0	262	0	1	1	1044	593 827
4-0130	0	282	0	1	0	1030	576 335
4-0200	0	292	0	1	1	971	548 328
4-0230	0	244	0	1	1	1010	522 243
4-0300	0	235	0	1	1	965	498 447
4-0330	0	231	0	1	1	956	482 321
4-0400	0	266	0	1	1	953	463 541
4-0430	0	251	0	1	1	932	448 161
4-0500	0	243	0	1	1	907	424 710
4-0530	0	244	0	1	1	916	391 595
4-0600	0	196	0	1	1	993	342 883

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
4-0630	0	197	0	1	1	907	332 101
4-0700	0	202	0	1	1	930	368 170
4-0730	0	176	0	1	1	924	367 137
4-0800	0	216	0	1	1	1004	353 137
4-0830	0	210	0	1	0	984	311 370
4-0900	0	229	0	1	1	906	206 669
4-0930	0	181	0	1	1	1021	117 741
4-1000	0	224	0	1	1	1085	−37 321
4-1030	0	201	0	1	1	1099	−123 817
4-1100	0	210	0	1	1	1285	−231 631
4-1130	0	178	0	1	1	1182	−304 191
4-1200	0	190	0	1	1	1406	−422 774
4-1230	0	212	0	1	1	1360	−511 622
4-1300	0	238	0	1	1	1395	−613 531
4-1330	0	234	0	1	1	1449	−671 400
4-1400	0	242	0	1	1	1431	−732 058
4-1430	0	328	0	1	1	1736	−769 397
4-1500	0	390	0	1	0	1824	−823 725
4-1530	44	22 240	0	2	1	1676	−842 444
4-1600	43	44 864	0	2	1	1640	−869 657
4-1630	0	23 692	0	2	0	1601	−887 273
4-1700	51	42 975	0	2	1	2279	−940 434
4-1730	69	66 188	0	2	1	1902	−987 021
4-1800	69	62 633	0	2	0	2555	−1 032 249
4-1830	0	46 495	0	2	1	2229	−1 050 887
4-1900	69	44 269	0	2	1	2548	−1 071 872
4-1930	68	66 610	0	2	1	2728	−1 079 959
4-2000	0	24 054	0	2	0	2866	−1 089 419
4-2030	85	44 188	0	2	1	2853	−1 097 059
4-2100	0	45 147	0	2	1	2644	−1 095 365
4-2130	40	44 173	0	2	1	2736	−1 083 522
4-2200	0	48 718	0	2	0	3149	−1 074 838
4-2230	44	44 513	0	2	1	3185	−1 060 043
4-2300	65	44 095	0	2	1	3290	−1 044 514
4-2330	64	44 127	0	2	1	3351	−1 026 087

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
6-0000	0	280	0	2	1	2098	−168 101
6-0030	0	293	0	2	1	2140	−125 703
6-0100	0	291	0	2	1	2354	−125 703
6-0130	0	474	0	2	1	1879	−85 628
6-0200	0	418	0	2	1	2362	−85 628
6-0230	0	621	0	2	1	1766	−893
6-0300	0	803	0	2	1	2451	−893
6-0330	0	279	0	2	0	2312	64 965
6-0400	0	269	0	2	0	1959	64 965
6-0430	0	563	0	2	1	1978	70 211
6-0500	0	258	0	2	0	2087	70 211
6-0530	0	272	0	2	0	2476	106 844
6-0600	0	441	0	2	1	1990	106 844
6-0630	0	279	0	2	1	1934	123 335
6-0700	0	270	0	2	0	2303	147 573
6-0730	0	242	0	2	1	2119	215 106
6-0800	0	359	0	2	0	2265	215 106
6-0830	0	446	0	2	1	2126	269 142
6-0900	0	261	0	2	1	2077	269 142
6-0930	0	366	0	2	1	2073	348 771
6-1000	0	515	0	2	1	1876	348 771
6-1030	0	762	0	2	1	1948	340 505
6-1100	0	548	0	2	0	2114	340 505
6-1130	0	304	0	2	1	2063	325 773
6-1200	0	30 848	0	3	1	1966	309 076
6-1230	42	22 296	0	3	2	1876	297 295
6-1300	0	45 888	0	3	0	1757	281 896
6-1330	0	42 231	0	3	1	1728	270 218
6-1400	0	41 885	0	3	0	1810	260 492
6-1430	42	65 202	0	3	1	1823	255 025
6-1500	0	14 643	0	3	0	1755	269 484
6-1530	0	402	0	1	0	1736	294 761
6-1600	46	44 177	0	3	1	1572	337 288
6-1630	46	22 008	0	3	1	1674	358 072
6-1700	46	21 980	0	3	2	1917	381 247
6-1730	0	3250	0	2	0	1988	394 147
6-1800	46	47 123	0	3	0	1634	422 147

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
6-1830	0	1136	0	2	1	1940	438 785
6-1900	0	1922	0	2	1	1906	462 488
6-1930	42	45 442	0	3	2	2185	470 473
6-2000	39	44 537	0	4	2	1597	482 749
6-2030	46	44 034	0	3	2	1649	495 675
6-2100	46	22 333	0	3	2	1691	515 946
6-2130	46	23 659	0	3	2	1779	531 204
6-2200	6	21 994	0	3	2	1767	552 744
6-2230	45	22 102	0	3	2	1990	563 174
6-2300	5	66 090	0	3	0	1905	579 301
6-2330	46	22 245	0	3	1	1841	583 580
7-0000	4	86 644	0	3	0	1757	579 605
7-0030	46	22 215	0	3	1	1870	563 881
7-0100	6	44 255	0	3	1	1851	528 849
7-0130	44	22 223	0	3	1	1940	500 177
7-0200	44	22 222	0	3	2	1967	460 889
7-0230	46	22 757	0	3	1	1985	451 006
7-0300	45	22 101	0	3	1	1809	440 883
7-0330	46	22 203	0	3	2	1792	438 928
7-0400	45	43 839	0	3	1	1749	446 867
7-0430	46	44 215	0	3	2	1750	470 411
7-0500	46	44 129	0	3	0	1922	511 560
7-0530	46	22 225	0	3	2	1963	522 933
7-0600	43	23 815	0	3	2	1848	542 113
7-0630	0	852	0	2	0	2214	538 961
7-0700	46	31 810	0	3	1	1879	532 849
7-0730	45	44 345	0	3	2	1828	533 304
7-0800	45	44 430	0	3	1	1868	533 816
7-0830	44	41 236	0	3	2	1838	534 017
7-0900	0	54 990	0	3	1	2155	541 202
7-0930	44	48 971	0	3	2	1817	542 968
7-1000	46	86 607	0	3	1	1719	558 647
7-1030	36	86 480	0	3	1	1873	591 341
7-1100	58	86 690	0	4	2	1689	623 877
7-1130	58	66 363	0	4	2	1847	633 294
7-1200	58	66 631	0	4	2	1847	647 838

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
7-1230	58	66 984	0	4	1	1781	638 318
7-1300	58	79 312	0	4	2	1826	617 415
7-1330	58	72 945	0	4	2	1829	612 553
7-1400	58	55 228	0	4	2	1745	582 059
7-1430	0	23 890	0	3	2	1934	572 114
7-1500	0	30 252	0	3	2	1870	541 467
7-1530	0	32 967	0	3	2	1824	520 202
7-1600	0	46 125	0	3	2	1964	462 470
7-1630	0	25 063	0	3	2	1874	427 210
7-1700	0	15 098	0	3	2	1864	383 423
7-1730	0	11 501	0	3	2	1879	370 414
7-1800	0	20 866	0	3	2	1825	355 831
7-1830	31	40 359	0	3	2	1795	345 176
7-1900	0	324	0	2	1	1680	330 505
7-1930	0	276	0	2	1	1641	325 703
7-2000	0	271	0	2	1	1975	316 659
7-2030	0	270	0	2	1	1552	315 644
7-2100	0	256	0	2	1	1962	323 383
7-2130	0	241	0	2	0	1989	330 707
7-2200	0	249	0	2	1	1534	341 571
7-2230	0	278	0	2	1	1585	353 935
7-2300	0	229	0	2	1	1634	374 228
7-2330	0	295	0	2	0	1616	389 140
8-0000	0	238	0	2	0	1731	404 080
8-0030	0	211	0	2	0	1507	411 925
8-0100	0	201	0	2	0	1489	443 340
8-0130	0	232	0	2	0	1387	451 453
8-0200	0	195	0	2	1	1634	451 809
8-0230	0	201	0	2	0	1399	464 515
8-0300	0	198	0	2	1	1626	493 553
8-0330	0	192	0	2	1	1424	501 667
8-0400	0	275	0	2	1	1623	499 744
8-0430	0	197	0	2	1	1446	503 261
8-0500	0	237	0	2	0	1619	515 749
8-0530	0	300	0	2	1	1640	521 780
8-0600	0	203	0	2	1	1640	535 867

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
8-0630	0	218	0	2	0	1410	534 509
8-0700	0	295	0	2	0	1427	574 149
8-0730	0	284	0	2	1	1334	594 440
8-0800	0	267	0	2	1	1533	623 264
8-0830	36	22 168	0	3	1	1419	638 968
8-0900	35	22 226	0	3	0	1580	662 791
8-0930	36	22 444	0	3	2	1403	687 248
8-1000	46	43 603	0	3	1	1184	732 672
8-1030	0	40 357	0	3	0	1251	766 761
8-1100	57	44 027	0	4	2	1138	841 468
8-1130	30	86 635	9580	4	2	1049	920 051
8-1200	80	69 649	106 591	4	3	1212	1 062 235
8-1230	88	86 639	230 899	4	3	953	1 161 941
8-1300	61	47 405	469 892	4	1	1259	1 296 826
8-1330	58	69 870	433 345	3	2	1020	1 357 700
8-1400	46	47 698	427 080	3	2	1527	1 445 475
8-1430	40	47 958	509 612	3	1	1551	1 489 579
8-1500	33	69 924	536 804	4	2	1130	1 539 617
8-1530	37	48 394	523 854	4	2	1642	1 569 065
8-1600	37	46 845	578 781	4	2	1267	1 613 890
8-1630	33	63 612	586 068	5	3	1155	1 633 645
8-1700	64	47 598	670 759	5	4	917	1 644 957
8-1730	37	67 423	576 504	4	2	1116	1 643 884
8-1800	34	47 338	681 299	4	3	1120	1 643 781
8-1830	80	47 298	590 654	4	2	993	1 639 362
8-1900	80	41 964	628 313	3	2	898	1 631 411
8-1930	80	47 132	660 233	5	4	1119	1 622 010
8-2000	97	45 789	709 573	3	1	1377	1 612 670
8-2030	44	54 612	545 477	4	2	673	1 593 477
8-2100	36	75 692	481 187	4	1	719	1 558 237
8-2130	37	86 453	476 630	4	3	671	1 538 186
8-2200	36	56 490	477 184	3	3	670	1 517 949
8-2230	49	46 611	440 183	3	1	1048	1 490 478
8-2300	42	69 259	396 320	4	2	667	1 438 243
8-2330	40	86 660	371 803	4	1	893	1 409 241

Instance	Gap	Time	Slack	S	S+F	Energy	Imbalance
9-0000	45	75 274	353 447	4	3	978	1 372 108
9-0030	48	69 520	332 147	4	2	826	1 360 573
9-0100	51	86 724	295 386	4	2	903	1 335 136
9-0130	61	68 756	286 432	4	3	952	1 319 586
9-0200	82	46 268	304 288	3	3	686	1 295 948
9-0230	89	46 293	347 916	3	2	735	1 272 133
9-0300	84	46 614	340 428	3	2	744	1 232 558
9-0330	89	86 616	201 520	4	1	715	1 207 653
9-0400	82	86 614	154 683	4	2	732	1 172 457
9-0430	75	45 985	274 165	5	5	762	1 147 827
9-0500	77	86 580	94 797	7	6	787	1 110 494
9-0530	83	68 959	114 911	3	3	749	1 087 135
9-0600	75	86 436	8317	3	2	689	1 050 451
9-0630	54	86 439	0	2	2	701	1 019 131
9-0700	70	86 520	0	3	2	789	975 895
9-0730	67	86 604	0	3	0	785	942 052
9-0800	66	39 378	0	3	2	737	891 820
9-0830	66	23 378	0	3	2	662	858 214
9-0900	66	54 855	0	3	2	682	809 500
9-0930	0	26 597	0	2	0	706	771 985

A.7. Results for H2-HC-EQ-B-P

The columns of the following tables contain the detailed results of the computational experiments for test set H2-HC-EQ-B-P described and discussed in Section 3.10. The first column contains the instance name. It consists of the virtual day, and the time in hours and minutes of the corresponding initial state, i.e., 2-0400 is the instance having the initial state from 4AM of virtual day 2. The second column states the gap of the MIP solve in % that was performed last. The third column states the runtime of the complete Algorithm 4. The fourth column states the boundary flow slack in kg applied in the final solution. While the fifth column denotes the total number of conducted simple state changes, the sixth column counts only those for which a flow direction change was simultaneously performed in the same network station and in the same time step. The last column states the amount of used compression energy in MWh, which we calculated a posteriori using the exact nonlinear power equation for turbo compressor units (P).

Instance	Gap	Time	Slack	S	S+F	Energy
1-1200	0	358	0	2	1	1761
1-1230	0	261	0	1	1	1990
1-1300	0	271	0	2	1	1667
1-1330	0	335	0	2	1	1657
1-1400	0	428	0	2	1	1702
1-1430	0	298	0	2	1	1807
1-1500	0	394	0	1	1	1702
1-1530	0	324	0	2	1	1754
1-1600	0	2326	0	2	1	1611
1-1630	0	1942	0	2	1	1576
1-1700	0	2918	0	2	1	1977
1-1730	0	3443	0	2	0	1984
1-1800	0	228	0	2	1	2308
1-1830	0	224	0	2	1	2401
1-1900	0	199	0	2	0	2144
1-1930	0	220	0	2	1	2091
1-2000	0	283	0	2	1	1879
1-2030	0	240	0	2	1	2217
1-2100	0	223	0	2	1	1932
1-2130	0	226	0	2	1	2189
1-2200	0	222	0	2	1	1812
1-2230	0	229	0	2	1	1814
1-2300	0	252	0	2	1	1735
1-2330	0	211	0	2	0	1773

Instance	Gap	Time	Slack	S	S+F	Energy
2-0000	0	220	0	2	0	1915
2-0030	0	229	0	2	1	1791
2-0100	0	228	0	2	1	1812
2-0130	0	215	0	2	1	2051
2-0200	0	243	0	2	1	1788
2-0230	0	218	0	2	1	1855
2-0300	0	216	0	2	1	1709
2-0330	0	227	0	2	1	1915
2-0400	0	219	0	2	1	1716
2-0430	0	217	0	2	1	1819
2-0500	0	198	0	2	1	1779
2-0530	0	218	0	2	1	1681
2-0600	0	231	0	2	0	1717
2-0630	0	225	0	2	1	1867
2-0700	0	1122	0	3	2	1684
2-0730	0	965	0	3	2	1798
2-0800	0	1438	0	3	2	1712
2-0830	0	494	0	3	1	1601
2-0900	0	614	0	3	1	1856
2-0930	0	1005	0	3	2	1902
2-1000	0	881	0	3	1	1778
2-1030	0	458	0	3	1	1728
2-1100	0	388	0	3	1	1587
2-1130	0	788	0	3	2	1611
2-1200	0	643	0	3	2	1724
2-1230	0	1140	0	3	2	1763
2-1300	0	678	0	3	2	1574
2-1330	0	2752	0	3	2	1646
2-1400	0	569	0	3	2	1794
2-1430	0	370	0	3	2	1704
2-1500	0	594	0	3	2	1687
2-1530	0	886	0	3	2	1536
2-1600	0	410	0	3	2	1402
2-1630	0	394	0	3	2	2240
2-1700	0	810	0	3	2	1838
2-1730	0	2563	0	3	1	1444
2-1800	0	460	0	3	2	1405

Instance	Gap	Time	Slack	S	S+F	Energy
2-1830	0	359	0	3	2	1353
2-1900	0	477	0	3	2	1349
2-1930	0	557	0	3	2	1258
2-2000	0	346	0	3	2	1137
2-2030	0	436	0	3	1	1247
2-2100	0	851	0	3	2	1173
2-2130	0	739	0	3	2	1205
2-2200	0	578	0	3	2	1157
2-2230	0	522	0	3	2	1192
2-2300	0	323	0	3	2	1114
2-2330	0	536	0	3	2	1113
3-0000	0	556	0	3	2	1182
3-0030	0	683	0	3	2	1087
3-0100	0	464	0	3	2	980
3-0130	0	441	0	3	1	1074
3-0200	0	490	0	3	2	998
3-0230	0	396	0	3	2	1087
3-0300	0	432	0	3	1	1081
3-0330	0	358	0	3	2	1045
3-0400	0	350	0	3	2	1054
3-0430	0	466	0	3	2	1124
3-0500	0	446	0	3	2	1107
3-0530	0	355	0	3	2	1322
3-0600	0	360	0	3	2	915
3-0630	0	290	0	3	1	1038
3-0700	0	293	0	3	2	1073
3-0730	0	718	0	3	2	1190
3-0800	0	635	0	3	2	920
3-0830	0	633	0	3	2	920
3-0900	0	147	0	1	0	1050
3-0930	0	153	0	1	1	995
3-1000	0	148	0	1	1	972
3-1030	0	199	0	2	2	1153
3-1100	0	180	0	1	1	1359
3-1130	0	149	0	1	1	1086
3-1200	0	162	0	1	1	1066

Instance	Gap	Time	Slack	S	S+F	Energy
3-1230	0	157	0	1	1	1172
3-1300	0	176	0	1	0	1283
3-1330	0	189	0	1	1	1356
3-1400	0	185	0	1	1	1328
3-1430	0	177	0	1	1	1340
3-1500	0	179	0	1	1	1437
3-1530	0	186	0	1	1	1275
3-1600	0	177	0	1	1	1275
3-1630	0	187	0	1	1	1275
3-1700	0	181	0	1	1	1317
3-1730	0	177	0	1	0	1264
3-1800	0	193	0	1	1	1214
3-1830	0	166	0	1	1	1219
3-1900	0	192	0	1	1	1289
3-1930	0	185	0	1	1	1256
3-2000	0	180	0	1	1	1175
3-2030	0	194	0	1	1	1582
3-2100	0	236	0	1	1	1555
3-2130	0	199	0	1	1	1401
3-2200	0	182	0	1	1	1457
3-2230	0	168	0	1	1	1614
3-2300	0	189	0	1	1	1648
3-2330	0	179	0	1	1	1460
4-0000	0	185	0	1	0	1391
4-0030	0	208	0	1	1	1375
4-0100	0	316	0	1	1	1526
4-0130	0	197	0	1	1	1351
4-0200	0	197	0	1	1	1217
4-0230	0	184	0	1	1	1117
4-0300	0	202	0	1	1	1110
4-0330	0	202	0	1	1	1154
4-0400	0	216	0	1	1	1084
4-0430	0	191	0	1	1	1047
4-0500	0	202	0	1	1	986
4-0530	0	223	0	1	1	1052
4-0600	0	213	0	1	0	961

Instance	Gap	Time	Slack	S	S+F	Energy
4-0630	0	221	0	1	0	1028
4-0700	0	506	0	1	1	1073
4-0730	0	163	0	1	1	1056
4-0800	0	364	0	1	1	1036
4-0830	0	205	0	1	1	1281
4-0900	0	240	0	1	1	1141
4-0930	0	192	0	1	1	1090
4-1000	0	195	0	1	1	1329
4-1030	0	166	0	1	1	1125
4-1100	0	222	0	1	1	1062
4-1130	0	159	0	1	1	996
4-1200	0	227	0	1	1	1115
4-1230	0	223	0	1	1	1113
4-1300	0	245	0	1	1	1188
4-1330	0	272	0	1	1	1145
4-1400	44	22 014	0	2	1	1189
4-1430	0	10 781	0	2	1	1226
4-1500	0	290	0	1	1	1275
4-1530	0	325	0	1	1	1233
4-1600	0	3402	0	2	0	1389
4-1630	0	1950	0	1	1	1345
4-1700	0	259	0	1	1	1341
4-1730	0	282	0	1	1	1397
4-1800	0	1051	0	2	2	1562
4-1830	0	383	0	1	1	1395
4-1900	0	269	0	1	1	1429
4-1930	0	219	0	1	0	1190
4-2000	0	221	0	1	1	1267
4-2030	0	221	0	1	0	1678
4-2100	0	259	0	1	0	1211
4-2130	0	248	0	1	0	1582
4-2200	0	246	0	1	0	1379
4-2230	0	234	0	1	1	1624
4-2300	0	233	0	1	0	1679
4-2330	0	249	0	1	1	1729

Instance	Gap	Time	Slack	S	S+F	Energy
6-0000	0	252	0	2	1	2352
6-0030	0	306	0	2	1	1521
6-0100	0	261	0	2	1	2177
6-0130	0	363	0	2	0	1757
6-0200	0	270	0	2	1	2208
6-0230	0	784	0	2	1	1777
6-0300	0	748	0	2	1	1893
6-0330	0	310	0	2	1	2371
6-0400	0	301	0	2	1	2299
6-0430	0	313	0	2	1	2257
6-0500	0	281	0	2	1	2098
6-0530	0	318	0	2	1	2203
6-0600	0	414	0	2	1	2491
6-0630	0	250	0	2	0	2303
6-0700	0	241	0	2	1	2108
6-0730	0	267	0	2	1	2214
6-0800	0	247	0	2	0	2272
6-0830	0	250	0	2	1	2375
6-0900	0	226	0	2	1	2547
6-0930	0	253	0	2	1	2350
6-1000	0	280	0	2	1	2522
6-1030	0	225	0	2	1	2043
6-1100	0	264	0	2	1	2366
6-1130	0	262	0	2	1	2408
6-1200	0	31 742	0	3	1	1899
6-1230	0	3473	0	3	1	1793
6-1300	0	6614	0	3	1	1642
6-1330	0	6664	0	3	1	1988
6-1400	0	10 623	0	3	0	1949
6-1430	0	7103	0	3	1	1717
6-1500	42	40 091	0	3	1	1760
6-1530	0	226	0	1	0	1792
6-1600	0	217	0	2	0	2021
6-1630	0	201	0	2	1	1657
6-1700	0	286	0	2	1	1755
6-1730	0	232	0	2	1	1796
6-1800	0	572	0	2	1	1877

Instance	Gap	Time	Slack	S	S+F	Energy
6-1830	0	233	0	2	1	2481
6-1900	0	307	0	2	0	2239
6-1930	0	249	0	2	0	2403
6-2000	0	298	0	3	0	2468
6-2030	0	13 694	0	2	1	1843
6-2100	0	7749	0	2	1	2665
6-2130	0	946	0	2	1	1883
6-2200	0	1448	0	2	1	2721
6-2230	0	1253	0	2	0	1865
6-2300	0	1195	0	2	1	2025
6-2330	0	1074	0	2	1	1918
7-0000	0	828	0	2	1	2218
7-0030	0	1524	0	2	1	2002
7-0100	0	1377	0	2	1	2550
7-0130	0	1570	0	2	0	1898
7-0200	0	1151	0	2	1	1907
7-0230	0	2303	0	2	0	2761
7-0300	0	5597	0	2	0	1903
7-0330	0	1299	0	2	1	1973
7-0400	0	4019	0	2	1	1879
7-0430	0	2124	0	2	1	2160
7-0500	0	1020	0	2	1	2294
7-0530	0	1025	0	2	1	2123
7-0600	0	1326	0	2	0	1987
7-0630	0	1726	0	2	0	2104
7-0700	0	716	0	2	0	2150
7-0730	0	954	0	2	1	2058
7-0800	0	696	0	2	1	2085
7-0830	0	917	0	2	0	2254
7-0900	0	852	0	2	1	1955
7-0930	0	725	0	2	1	1930
7-1000	0	837	0	2	1	2307
7-1030	0	615	0	2	1	1831
7-1100	0	3578	0	2	1	2324
7-1130	0	403	0	2	1	2455
7-1200	0	1140	0	2	1	2781

Instance	Gap	Time	Slack	S	S+F	Energy
7-1230	0	339	0	2	1	1919
7-1300	0	724	0	2	1	2246
7-1330	0	991	0	2	1	1886
7-1400	0	477	0	2	1	2075
7-1430	0	814	0	2	0	1881
7-1500	0	516	0	2	1	1591
7-1530	0	1026	0	2	1	1675
7-1600	0	710	0	2	0	1714
7-1630	0	261	0	2	0	2194
7-1700	0	243	0	2	1	2370
7-1730	0	236	0	2	1	2498
7-1800	0	295	0	2	0	1755
7-1830	0	282	0	2	0	1647
7-1900	0	267	0	2	1	1977
7-1930	0	222	0	2	1	2422
7-2000	0	236	0	2	0	2354
7-2030	0	292	0	2	1	2000
7-2100	0	292	0	2	0	2095
7-2130	0	201	0	2	1	1896
7-2200	0	322	0	2	1	1846
7-2230	0	261	0	2	1	1777
7-2300	0	210	0	2	0	1783
7-2330	0	245	0	2	1	1906
8-0000	0	199	0	2	0	1760
8-0030	0	202	0	2	0	1703
8-0100	0	385	0	2	0	1802
8-0130	0	376	0	2	0	1862
8-0200	0	378	0	2	1	1739
8-0230	0	364	0	2	0	1646
8-0300	0	307	0	2	1	2500
8-0330	0	297	0	2	0	1936
8-0400	0	268	0	2	0	2008
8-0430	0	348	0	2	0	2380
8-0500	0	383	0	2	1	1905
8-0530	0	337	0	2	1	2008
8-0600	0	458	0	2	0	1663

Instance	Gap	Time	Slack	S	S+F	Energy
8-0630	0	336	0	2	1	2315
8-0700	0	332	0	2	1	2090
8-0730	0	329	0	2	0	1685
8-0800	0	431	0	2	1	1655
8-0830	0	328	0	2	0	2181
8-0900	0	422	0	2	1	1713
8-0930	0	625	0	2	1	2218
8-1000	0	416	0	2	1	1926
8-1030	0	390	0	2	1	1560
8-1100	0	398	0	2	0	1875
8-1130	0	323	0	2	0	2093
8-1200	0	394	0	2	0	1838
8-1230	0	316	0	2	1	1697
8-1300	0	332	0	2	0	2191
8-1330	0	441	0	2	1	1790
8-1400	0	350	0	2	1	2032
8-1430	0	318	0	2	1	2020
8-1500	0	394	0	2	1	1872
8-1530	0	423	0	2	1	1899
8-1600	0	379	0	2	1	1843
8-1630	0	624	0	2	1	1795
8-1700	0	431	0	2	1	1712
8-1730	0	435	0	2	1	1676
8-1800	0	401	0	2	1	1742
8-1830	0	305	0	2	1	1692
8-1900	0	304	0	1	1	1544
8-1930	0	262	0	1	1	1938
8-2000	0	271	0	1	1	1732
8-2030	0	267	0	1	0	1782
8-2100	0	312	0	1	1	1490
8-2130	0	271	0	1	1	1665
8-2200	0	262	0	1	1	1433
8-2230	0	283	0	1	0	1440
8-2300	0	312	0	1	1	1439
8-2330	0	302	0	1	1	1522

Instance	Gap	Time	Slack	S	S+F	Energy
9-0000	0	336	0	1	1	1444
9-0030	0	268	0	1	0	1112
9-0100	0	301	0	1	0	1413
9-0130	0	289	0	1	1	1275
9-0200	0	233	0	1	1	967
9-0230	0	293	0	1	1	1266
9-0300	0	309	0	1	1	1339
9-0330	0	286	0	1	1	1073
9-0400	0	294	0	1	1	1247
9-0430	0	306	0	1	0	1229
9-0500	0	332	0	1	1	1222
9-0530	0	247	0	1	1	914
9-0600	0	288	0	1	1	1167
9-0630	0	284	0	1	1	1024
9-0700	0	287	0	1	1	1149
9-0730	0	265	0	1	0	1119
9-0800	0	278	0	1	1	1160
9-0830	0	346	0	1	1	1128
9-0900	0	300	0	1	1	1103
9-0930	0	1114	0	1	1	973

B. Appendix to Chapter 4

B.1. The Maximum Minimum Cost Flow Problem

This section discusses the Maximum Min-Cost-Flow Problem (MaxMCF), a natural generalization of the Maximum Transportation Problem (MaxTP) introduced in Section 4.2. It is equivalent to MaxTP, but we consider additional capacity restrictions on the arcs. In particular, for MaxMCF, we are given capacity values c_a for each $a \in \mathcal{A}$, which the arc flows f_a are not allowed to exceed in any feasible solution, see Section 2.3.1 for more information regarding MCF. Thus, an instance of MaxMCF is a sextuple $\mathcal{I} = (\mathcal{V}, \mathcal{A}, \ell, c, \underline{b}, \bar{b})$. Accordingly, the goal of MaxMCF is to find an admissible supply and demand vector such that the optimal objective value of the induced MCF instance is maximized. Again, we assume that there exists an uncapacitated directed path from each entry towards each exit to ensure the existence of a feasible solution for all MCF instances induced by admissible supply and demand vectors. The bilevel model for MaxTP from Section 4.2.4 can be adapted as follows.

$$\max_{b} \quad \sum_{a \in \mathcal{A}} \ell_a f_a \tag{B.1}$$

$$\text{s.t.} \quad \sum_{u \in \mathcal{V}^+} b_u + \sum_{w \in \mathcal{V}^-} b_w = 0 \tag{B.2}$$

$$b_v \in [\underline{b}_v, \bar{b}_v] \qquad \forall v \in \mathcal{V}^+ \cup \mathcal{V}^- \tag{B.3}$$

$$\min_{f} \quad \sum_{a \in \mathcal{A}} \ell_a f_a \tag{B.4}$$

$$\text{s.t.} \sum_{a \in \delta^+(v)} f_a - \sum_{a \in \delta^-(v)} f_a = b_v \qquad \forall v \in \mathcal{V}^+ \cup \mathcal{V}^- \tag{B.5}$$

$$\sum_{a \in \delta^+(v)} f_a - \sum_{a \in \delta^-(v)} f_a = 0 \qquad \forall v \in \mathcal{V}^0 \tag{B.6}$$

$$f_a \in [0, c_a] \quad \forall a \in \mathcal{A} \tag{B.7}$$

Since MaxMCF is a generalization of MaxTP, its NP-hardness follows directly from Theorem 1 in Section 4.2.3. However, due to the capacity constraints, we conjecture that an even stronger statement holds. The complicating fact that Lemma 1 regarding the structure of an optimal solution does not hold for MaxMCF, namely that there exists one with a bound-close supply and demand vector, can be seen as evidence for our claim. A counterexample demonstrating this is shown in Figure B.1.

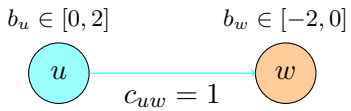


Figure B.1.: Counterexample demonstrating that Lemma 1 regarding the bound-closeness of an optimal solution for MaxTP does not hold for MaxMCF. The unique optimal solution for the example instance shown here is $b_u = 1$, $b_w = -1$, and $f_{uw} = 1$.

Conjecture 1. MaxMCF *is APX-hard.*

Although we have not proven Conjecture 1 yet, we can show the same statement for a different variant of MaxMCF. As mentioned, we assume that there exists an uncapacitated directed path from each entry towards each exit in the network, guaranteeing the existence of a feasible solution for all MCF instances induced by admissible supply and demand vectors. In the following, we consider MaxMCF with this connectedness-condition being dropped, which we denote by MaxMCF-CC. The goal of MaxMCF-CC is to find an admissible supply and demand vector b for which the induced MCF instance admits a feasible solution and for which the objective value of an optimal solution is maximized. Note that the bilevel optimization model presented above remains valid since the leader must ensure that the lower level, i.e., the MCF instance, admits a feasible solution by definition. Nevertheless, we once more note that there may now exist admissible supply and demand vectors that induce infeasible MCF instances and are therefore not chosen by the leader.

Theorem 4. MaxMCF-CC *is APX-hard.*

We reduce from the Maximum Independent Set with Bounded Degree Problem (MIS). Its definition is adapted from GT23 in Ausiello et al. [7]:

Definition 12. *Let $H = (W, E)$ be an undirected graph such that the degree of each node is bounded by some constant $B \geq 3$, i.e., we have $\delta(v) \leq B$ for all $v \in W$. The goal of the* Maximum Independent Set with Bounded Degree Problem *is to find a maximum subset $W' \subseteq W$ w.r.t. the cardinality such that no two vertices in W' are joined by an edge.*

Theorem 5. MIS *is APX-complete.*

Proof. A proof can be found in the papers of Berman and Fujito [13] and Papadimitriou and Yannakakis [125]. □

Given an undirected graph $H = (W, E)$ as instance of MIS, we create a corresponding MaxMCF-CC instance $\mathcal{I}_H$ as follows. Note that we assume w.l.o.g. that H contains no isolated vertices since these are trivially contained in every maximum independent set.

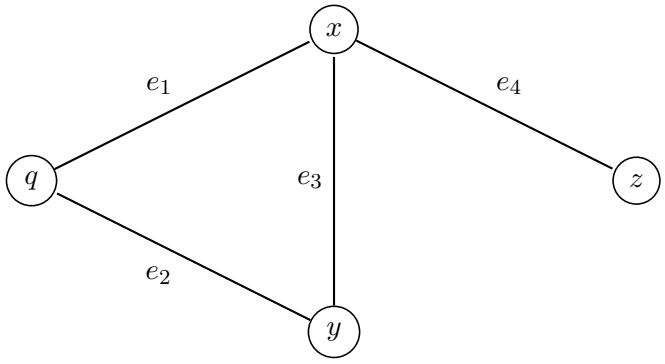


Figure B.2.: An example MIS instance $H = (W, E)$.

First, for each vertex $v \in W$, we add an entry $v^+ \in \mathcal{V}^+$ and two inner nodes $v^0, v^1 \in \mathcal{V}^0$. Furthermore, for each edge $e \in E$, we add an exit $e^- \in \mathcal{V}^-$. For each entry $v^+ \in \mathcal{V}^+$, we define $\underline{b}_{v^+} := 0$ as the lower supply bound while the upper supply bound is set to $\bar{b}_{v^+} := \delta(v)$, i.e., the degree of the corresponding vertex $v \in W$. Additionally, for each exit $e^- \in \mathcal{V}^-$, we define $\underline{b}_{e^-} := -1$ and $\bar{b}_{e^-} := 0$.

Next, we add four different types of arcs to $\mathcal{I}_H$, i.e., $\mathcal{A} := \mathcal{A}^1 \cup \mathcal{A}^2 \cup \mathcal{A}^3 \cup \mathcal{A}^4$. First, for each vertex $v \in W$, we add an arc from $v^+ \in \mathcal{V}^+$ to $v^0 \in \mathcal{V}^0$, i.e., $\mathcal{A}^1 := \{(v^+, v^0) \mid v \in W\}$. Second, for $v \in W$, we add an arc from $v^0 \in \mathcal{V}^0$ to each exit $e^- \in \mathcal{V}^-$ whose corresponding edge $e \in E$ is incident to v, i.e., $\mathcal{A}^2 := \{(v^0, e^-) \mid v \in W, e \in \delta(v)\}$. Third, for $v \in W$, we add an arc from $v^+ \in \mathcal{V}^+$ to $v^1 \in \mathcal{V}^0$, i.e., $\mathcal{A}^3 := \{(v^+, v^1) \mid v \in W\}$. And fourth, for each $v \in W$, we add an arc from $v^1 \in \mathcal{V}^+$ to each exit $e^- \in \mathcal{V}^-$ whose corresponding edge $e \in E$ is incident to v, i.e., $\mathcal{A}^4 := \{(v^1, e^-) \mid v \in W, e \in \delta(v)\}$. Additionally, we define $\ell_a := 0$ for each $a \in \mathcal{A}^1 \cup \mathcal{A}^2 \cup \mathcal{A}^4$ and $\ell_a := 1$ for each $a \in \mathcal{A}^3$. Finally, we set $c_a := \delta(v) - 1$ for all $a = (v^+, v^0) \in \mathcal{A}^1$ and $c_a := 1$ for all $a \in \mathcal{A}^2 \cup \mathcal{A}^3 \cup \mathcal{A}^4$. This concludes the construction of instance $\mathcal{I}_H$.

Note that $\mathcal{I}_H$ is of linear size w.r.t. H since the number of nodes is equal to $|\mathcal{V}| = 3|W|+|E|$ and the number of arcs is equal to $|\mathcal{A}| = 2|W|+4|E|$. The MaxMCF-CC instance $\mathcal{I}_H$ corresponding to the example MIS instance H in Figure B.2 is shown in Figure B.3.

Next, we introduce some definitions regarding a feasible solution (b, f) for $\mathcal{I}_H$. For each entry $v^+ \in \mathcal{V}^+$, the flow towards an exit $e^- \in \mathcal{V}^-$, whose corresponding edge $e \in E$ is incident to the corresponding node $v \in W$, can be uniquely partitioned into flow on two different paths: Flow on the short path $v^+ \to v^0 \to e^-$ having length 0, and flow on the long path $v^+ \to v^1 \to e^-$ with having length 1. The corresponding values can directly be read from $f_{v^0 e^-}$ and $f_{v^1 e^-}$, respectively.

Furthermore, we call a solution (b, f) assigning if it is feasible, and for each exit $e^- \in \mathcal{V}^-$ with $b_{e^-} < 0$, corresponding to an edge $e = \{x, y\} \in E$, we have that either $f_{x^0 e^-} + f_{x^1 e^-} > 0$ or $f_{y^0 e^-} + f_{y^1 e^-} > 0$ holds but not both.

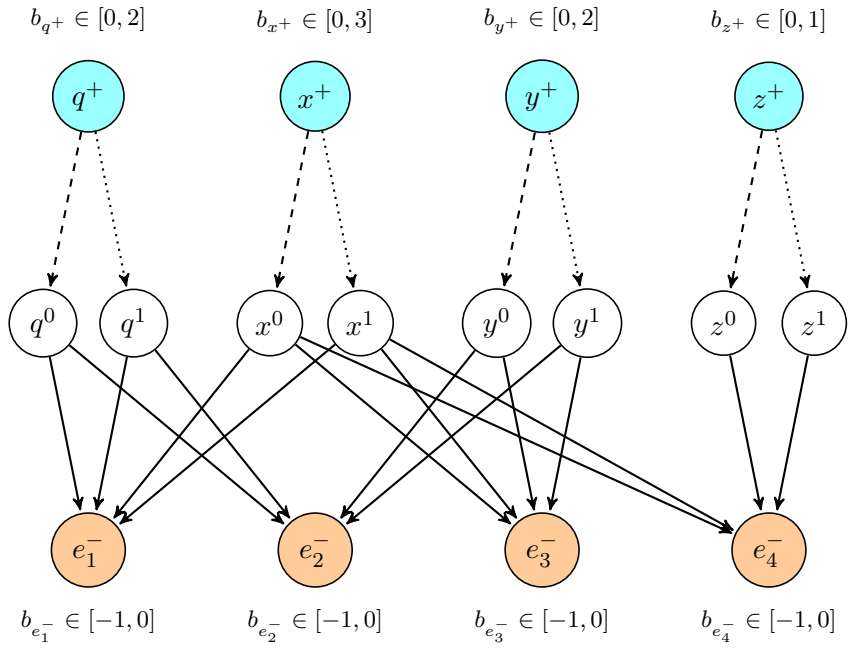


Figure B.3.: MaxMCF-CC instance $\mathcal{I}_H$ corresponding to MIS instance H from Figure B.2. For a dashed arc $a = (v^+, v^0) \in \mathcal{A}^1$, we have $c_a = \delta(v) - 1$ and $\ell_a = 0$, while for a dotted arc $a \in \mathcal{A}^3$, we have $c_a = 1$ and $\ell_a = 1$. Finally, for a solid arc $a \in \mathcal{A}^2 \cup \mathcal{A}^4$, we have $c_a = 1$ and $\ell_a = 0$. Note that the connectedness-condition described in Section 4.2 does not hold for this flow network.

Lemma 19. *Let (b, f) be a feasible solution for $\mathcal{I}_H$. There exists an assigning solution $(\tilde{b}, \tilde{f})$, which can be determined in $\mathcal{O}(|E|)$, such that $c(\tilde{b}, \tilde{f}) \geq c(b, f)$.*

Proof. Let (b, f) be a feasible solution and assume there exists an exit $e^- \in \mathcal{V}^-$ corresponding to $e = \{x, y\} \in E$ such that we have $f^e_x := f_{x^0e^-} + f_{x^1e^-} > 0$ and $f^e_y := f_{y^0e^-} + f_{y^1e^-} > 0$. W.l.o.g. we assume that $\bar{b}_{x^+} - b_{x^+} \leq \bar{b}_{y^+} - b_{y^+}$. Next, we shift the supply routed from y^+ towards e^- to x^+. Therefore, let $\Omega := c_{x^+x^0} - f_{x^+x^0}$ denote the remaining capacity on the short path from x^+ towards e^-. We define

$$\tilde{b}_v := \begin{cases} b_{x^+} + f^e_y & \text{for } v = x^+ \in \mathcal{V}^+ \\ b_{y^+} - f^e_y & \text{for } v = y^+ \in \mathcal{V}^+ \\ b_v & \text{otherwise} \end{cases}$$

and

$$\tilde{f}_a := \begin{cases} f_{x^+x^0} + \min\{\Omega, f^e_y\} & \text{if } a = (x^+, x^0) \\ f_{x^0e^-} + \min\{\Omega, f^e_y\} & \text{if } a = (x^0, e^-) \\ f_{x^+x^1} + \max\{f^e_y - \Omega, 0\} & \text{if } a = (x^+, x^1) \\ f_{x^1e^-} + \max\{f^e_y - \Omega, 0\} & \text{if } a = (x^1, e^-) \\ f_{y^+y^0} - f_{y^0e^-} & \text{if } a = (y^+, y^0) \\ 0 & \text{if } a = (y^0, e^-) \\ f_{y^+y^1} - f_{y^1e^-} & \text{if } a = (y^+, y^1) \\ 0 & \text{if } a = (y^1, e^-) \\ f_a & \text{otherwise.} \end{cases}$$

We know that x^+ can only supply exits whose corresponding edges are incident to $x \in W$. Thus, we have

$$\begin{aligned} b_{x^+} = f_{x^+x^0} + f_{x^+x^1} &= \sum_{\tilde{e} \in \delta(x)} f_{x^0\tilde{e}^-} + \sum_{\tilde{e} \in \delta(x)} f_{x^1\tilde{e}^-} = \sum_{\tilde{e} \in \delta(x)} (f_{x^0\tilde{e}^-} + f_{x^1\tilde{e}^-}) \\ &\leq \Big(\sum_{\tilde{e} \in \delta(x)} |b_{\tilde{e}^-}| \Big) - (f_{y^0e^-} + f_{y^1e^-}) \leq \delta(x) - f^e_y = \bar{b}_x - f^e_y, \end{aligned}$$

showing that $\tilde{b}_{x^+} = b_{x^+} + f^e_y \leq \bar{b}_x$. Hence, $\tilde{b}$ is an admissible supply and demand vector. Furthermore, while the flow on the short and long path from y^+ towards e^- is set to 0, we route up to Ω units from x^+ towards e^- on the short path, and the remaining supply on the long path.

If there was flow on the short path from y^+ towards e^- in f and if there is still flow on some long paths starting at y^+ in $\tilde{f}$, we need to shift it to the short paths to transform $\tilde{f}$ to an optimal solution for the induced MCF instance. This is done through Algorithm 12.

Finally, it remains to show that $c(\tilde{b}, \tilde{f}) \geq c(b, f)$. Recall that we assumed that $\bar{b}_{x^+} - b_{x^+} \leq \bar{b}_{y^+} - b_{y^+}$. There are two cases we have to consider: First, if $\bar{b}_{y^+} - b_{y^+} \leq 1$, then $\bar{b}_{x^+} - b_{x^+} \leq 1$, and all the shifted supply is routed along the long path from x^+

Algorithm 12: Shift flow to ensure the optimality of $\tilde{f}$

for $e \in \delta(y)$ **do**
  $r \leftarrow \min\{c_{y^+y^0} - \tilde{f}_{y^+y^0}, \tilde{f}_{y^1e^-}\}$
  $\tilde{f}_{y^+y^0} \leftarrow \tilde{f}_{y^+y^0} + r$
  $\tilde{f}_{y^0e^-} \leftarrow \tilde{f}_{y^0e^-} + r$
  $\tilde{f}_{y^+y^1} \leftarrow \tilde{f}_{y^+y^1} - r$
  $\tilde{f}_{y^1e^-} \leftarrow \tilde{f}_{y^1e^-} - r$

towards e^-. Second, if $\bar{b}_{y^+} - b_{y^+} \geq 1$, all the supply from y^+ towards e^- was routed along the short path in f. In both cases, we the objective value is not decreased, implying $c(\tilde{b}, \tilde{f}) \geq c(b, f)$.

Iteratively applying the procedure above to all exits e^- results in an assigning solution $(\tilde{b}, \tilde{f})$ such that $c(\tilde{b}, \tilde{f}) \geq c(b, f)$ in $\mathcal{O}(|E|)$. This is because for each exit $e^- \in \mathcal{V}^-$, the construction of $(\tilde{b}, \tilde{f})$ can be done in constant time. □

Next, let us call a solution (b, f) *bound-tight* if it is feasible, and if either $b_{v^+} = 0$ or $b_{v^+} = \delta(v)$ holds for all $v^+ \in \mathcal{V}^+$.

Lemma 20. *Let (b, f) be a feasible solution for $\mathcal{I}_H$. There exists a bound-tight solution $(\tilde{b}, \tilde{f})$, which can be determined in $\mathcal{O}(|V| + |E|)$, with $c(\tilde{b}, \tilde{f}) \geq c(b, f)$.*

Proof. Let (b, f) be a feasible solution. Using Lemma 19, w.l.o.g. we assume that (b, f) is assigning. We define

$$\tilde{b}_v := \begin{cases} \bar{b}_{x^+} & \text{if } v = x^+ \in \mathcal{V}^+ \text{ with } b_{x^+} > \delta(x) - 1 \\ -1 & \text{if } v = e^- \in \mathcal{V}^- \text{ with } e = \{x, y\} \text{ if } b_{x^+} > \delta(x) - 1 \text{ or } b_{y^+} > \delta(y) - 1 \\ 0 & \text{if } v = x^+ \in \mathcal{V}^+ \text{ with } b_{x^+} \leq \delta(x) - 1 \\ 0 & \text{if } v = e^- \in \mathcal{V}^- \text{ with } e = \{x, y\} \text{ if } b_{x^+} \leq \delta(x) - 1 \text{ and } b_{y^+} \leq \delta(y) - 1 \end{cases}$$

and

$$\tilde{f}_a := \begin{cases} \delta(x) - 1 & \text{if } a = x^+x^0 \text{ and } b_{x^+} > \delta(x) - 1 \\ 1 & \text{if } a = x^+x^1 \text{ and } b_{x^+} > \delta(x) - 1 \\ f_a & \text{if } a = x^0e^- \text{ and } b_{x^+} > \delta(x) - 1 \\ f_a + (1 - |b_{e^-}|) & \text{if } a = x^1e^- \text{ and } b_{x^+} > \delta(x) - 1 \\ 0 & \text{otherwise.} \end{cases}$$

First, the supply of each entry $x^+ \in \mathcal{V}^+$ with $b_{x^+} > \delta(x) - 1$ is increased up to its upper bound, i.e., $\tilde{b}_{x^+} := \bar{b}_{x^+} = \delta(x)$, and the demand of each exit e^- with $e \in \delta(x)$ up to -1, i.e., $\tilde{b}_{e^-} := -1$. The additional supply of $1 - |b_{e^-}|$ is then routed along the corresponding long path.

Second, the supply of all entries x^+ with $b_{x^+} \leq \delta(x) - 1$, the demands of their assigned exits, as well as the flows on the corresponding short paths are set to 0. Since $b_{x^+} \leq \delta(x) - 1$, there is no flow on any of the corresponding long paths.

In both cases, the objective value does not decrease and we determined a bound-tight solution $(\tilde{b}, \tilde{f})$ with $c(\tilde{b}, \tilde{f}) \geq c(b, f)$ in $\mathcal{O}(|V| + |E|)$. □

Lemma 21. *Let (b, f) be a bound-tight solution for* MaxMCF-CC *instance $\mathcal{I}_H$. Then $W' := \{v \in W \mid b_{v^+} = \bar{b}_{v^+}\}$ is an independent set in H. Furthermore, we have that $c(b, f) = |W'|$.*

Proof. Assume W' is not independent. Hence, there exist two nodes $x, y \in W'$ such that $b_{x^+} = \bar{b}_{x^+}$, $b_{y^+} = \bar{b}_{y^+}$, and $\{x, y\} = e \in E$. Since only exits whose corresponding edges are contained in $\delta(x) \cup \delta(y)$ can be supplied by x^+ and y^+, and because $e \in \delta(x) \cap \delta(y)$, it holds that

$$\bar{b}_{x^+} + \bar{b}_{y^+} = \sum_{e \in \delta(x) \cup \delta(y)} |b_{e^-}| \leq |\delta(x) \cup \delta(y)| \leq \delta(x) + \delta(y) - 1 < \bar{b}_{x^+} + \bar{b}_{y^+},$$

which is a contradiction. Thus, W' is an independent set and

$$\begin{aligned} c(b, f) &= \sum_{a \in \mathcal{A}^1} \ell_a f_a + \sum_{a \in \mathcal{A}^2} \ell_a f_a + \sum_{a \in \mathcal{A}^3} \ell_a f_a + \sum_{a \in \mathcal{A}^4} \ell_a f_a \\ &= \sum_{a \in \mathcal{A}^3} f_{v^+v^1} = \sum_{v \in W} f_{v^+v^1} = \sum_{v \in W'} 1 = |W'|. \end{aligned}$$

□

Lemma 22. *Let W' be an independent set in H. Then there exists a bound-tight solution (b, f) for $\mathcal{I}_H$ with $|W'| = c(b, f)$.*

Proof. Consider (b, f) defined as

$$b_v := \begin{cases} \delta(x) & \text{if } v = x^+ \in \mathcal{V}^+ \text{ and } x \in W' \\ -1 & \text{if } v = e^- \in \mathcal{V}^- \text{ and } e \in \delta(W') \\ 0 & \text{otherwise} \end{cases}$$

and

$$f_a := \begin{cases} \delta(x) - 1 & \text{if } a = x^+x^0 \text{ and } x \in W' \\ \frac{\delta(x)-1}{\delta(x)} & \text{if } a = x^0e^- \text{ and } x \in W' \\ 1 & \text{if } a = x^+x^1 \text{ and } x \in W' \\ \frac{1}{\delta(x)} & \text{if } a = x^1e^- \text{ and } x \in W' \\ 0 & \text{otherwise.} \end{cases}$$

By construction, (b, f) is feasible, bound-tight and we have $c(b, f) = |W'|$. □

Lemma 23. *There exists an independent set W' of H of size $|W'| = k$ if and only if there exists a bound-tight solution (b, f) for $\mathcal{I}_H$ with $c(b, f) = k$.*

Proof. Given a feasible and bound-tight solution (b, f) for $\mathcal{I}_H$ with $c(b, f) = k$, the induced independent set W' from Lemma 21 has size $|W'| = k$. Conversely, if there exists an independent set W' of H with size k, by Lemma 22 there exists a bound-tight solution (b, f) for $\mathcal{I}_H$ with $c(b, f) = k$. □

Corollary 12. *The size of a maximum independent set W' of H is $|W'| = k$ if and only if an optimal solution (b, f) for $\mathcal{I}_H$ has objective value $c(b, f) = k$.*

Using the previous results, we can now prove Theorem 4.

Theorem 4. MaxMCF-CC *is APX-hard.*

Proof. Suppose there exists a PTAS for MaxMCF-CC yielding a $(1 - \varepsilon)$-factor approximate solution. Let $\mathcal{I}_H = (V, A)$ be the corresponding MaxMCF-CC instance for MIS instance $H = (W, E)$ and let k denote the optimal objective value of $\mathcal{I}_H$, which is equal to the size of a maximum independent set in H by Corollary 12. A PTAS would give us a feasible solution (b, f) for $\mathcal{I}_H$ with solution value $c(b, f) \geq (1 - \varepsilon)k$. Using Lemma 19 and Lemma 20, we can determine a bound-tight feasible solution $(\tilde{b}, \tilde{f})$ in polynomial time w.r.t. the number of vertices and edges with $c(\tilde{b}, \tilde{f}) \geq c(b, f)$. Next, by Lemma 21 we can extract an independent set in $\mathcal{O}(|W|)$, which has size at least $(1 - \varepsilon)k$. Hence, a $(1 - \varepsilon)$-factor PTAS for MaxMCF-CC together with the algorithms in Lemma 19, Lemma 20, and Lemma 21 would yield a $(1 - \varepsilon)$-factor PTAS for MIS. Thus, no PTAS for MaxMCF-CC can exist unless P=NP. □

C. Appendix to Chapter 5

C.1. Detailed Computational Results for LCCP

This appendix contains the detailed results regarding the computational experiments described and analyzed in Section 5.7.

In Tables C.1 to C.4, we present the results of our experiments regarding the two MIP approaches for instances with up to 100 vertices. Here, the instance names are stated in the first column and incorporate the number of vertices, i.e., instance bayg29 features 29 vertices. For the ATSPLIB instances with the "ftv"-prefix, we must add one vertex, e.g., ftv33 has 34 vertices. Instance kro124p is an outlier w.r.t. this nomenclature and features 100 vertices. The second column shows the percentage of edges that were removed by the preprocessing routine. The size of the MCV heuristic solution is stated in column UB (MCV heur). Moreover, the lower bound obtained from a maximum clique in the conflict graph H_2 is shown in column LB (H_2-clique). For both MIP approaches, the upper and the lower bound at the end of the solving processes are stated, which was either reached when the problem was solved or when the time limit was hit, which is indicated by TL in the column denoting the respective run times.

In Tables C.5 to C.8, we present the results of the computational experiments regarding the MCV heuristic on large instances between 100 and 1000 vertices. Recall that we considered two versions of MCV. First, MCV Enhanced, where we applied the an TSP algorithm in an intermediate step and and thereby try to directly decrease the length of current cycle after its creation. Second, MCV Basic, where we omit this step. The instance names are again stated in the first column and incorporate the number of vertices, i.e., instance eil101 features 101 vertices. For the ATSPLIB instances with the "ftv"-prefix, we must add one vertex, i.e., ftv170 has 171 vertices. For both versions of the heuristic, we state the size of the obtained solutions in the SolVal column and their run times.

Table C.1.: Results for TSPLIB with critical weights from $[\frac{\tau^*}{8}, \frac{\tau^*}{4}]$

				MTZ			SEC		
Instance name	Removed edges in %	UB (MCV heur)	LB (H_2-clique)	UB	LB	time (sec)	UB	LB	time (sec)
burma14	73.6	6	6	6	6	1	6	6	1
ulysses16	54.2	6	5	6	6	1	6	6	1
gr17	70.6	8	7	8	8	1	8	8	1
gr21	67.6	8	6	8	8	4	8	8	3
ulysses22	55.0	7	6	7	7	4	7	7	2
gr24	62.7	7	7	7	7	1	7	7	1
fri26	57.8	8	6	8	8	6	8	8	10
bayg29	58.6	10	7	8	8	42	8	8	23
bays29	58.1	8	6	8	8	36	8	8	21
dantzig42	56.1	9	7	9	9	1552	9	9	575
swiss42	52.4	9	7	9	8	TL	9	9	284
att48	53.6	9	6	8	8	4563	8	8	2893
gr48	48.0	9	5	9	8	TL	9	8	TL
hk48	48.6	10	6	9	8	TL	9	9	14417
eil51	40.9	9	5	9	8	TL	9	9	19242
berlin52	38.8	9	6	9	9	2956	9	9	2260
brazil58	40.9	9	5	8	7	TL	9	7	TL
st70	44.5	10	5	10	7	TL	10	7	TL
eil76	27.4	10	5	10	6	TL	10	6	TL
pr76	32.5	10	6	10	7	TL	10	7	TL
gr96	34.8	10	5	10	6	TL	10	7	TL
rat99	34.5	10	5	10	6	TL	10	6	TL
kroA100	43.3	10	5	10	6	TL	10	5	TL
kroB100	40.0	10	5	10	6	TL	10	6	TL
kroC100	44.1	11	6	11	7	TL	11	6	TL
kroD100	39.6	11	6	11	6	TL	11	6	TL
kroE100	42.3	11	5	11	6	TL	11	6	TL
rd100	36.7	11	5	11	6	TL	11	6	TL

Table C.2.: Results for TSPLIB with critical weights from $[\frac{\tau^*}{6}, \frac{\tau^*}{2}]$

				MTZ			SEC		
Instance name	Removed edges in %	UB (MCV heur)	LB (H_2-clique)	UB	LB	time (sec)	UB	LB	time (sec)
burma14	44.0	5	4	5	5	1	5	5	1
ulysses16	32.5	4	4	4	4	1	4	4	1
gr17	44.9	5	4	5	5	1	5	5	1
gr21	41.0	6	3	5	5	2	5	5	1
ulysses22	28.6	5	4	5	5	4	5	5	1
gr24	20.7	6	4	5	5	11	5	5	3
fri26	32.3	6	5	6	6	87	6	6	9
bayg29	25.4	6	4	5	5	30	5	5	13
bays29	27.6	6	5	6	6	32	6	6	14
dantzig42	28.7	7	4	6	6	20686	6	6	15081
swiss42	18.6	7	4	6	6	13111	6	6	6833
att48	26.5	8	5	6	5	TL	6	6	4939
gr48	18.5	7	4	7	5	TL	7	6	TL
hk48	25.0	7	4	6	6	18843	6	6	3656
eil51	10.0	7	3	7	5	TL	7	5	TL
berlin52	19.9	7	4	7	6	TL	7	6	TL
brazil58	14.7	6	3	6	4	TL	6	5	TL
st70	9.0	7	3	7	5	TL	7	5	TL
eil76	6.5	7	3	7	5	TL	7	5	TL
pr76	7.3	7	3	7	5	TL	7	5	TL
gr96	8.9	7	3	7	5	TL	7	5	TL
rat99	12.1	7	3	7	5	TL	7	5	TL
kroA100	16.2	8	3	8	5	TL	8	5	TL
kroB100	13.0	8	3	8	5	TL	8	5	TL
kroC100	17.3	7	3	7	5	TL	7	5	TL
kroD100	12.8	7	3	7	5	TL	7	5	TL
kroE100	13.7	7	3	7	5	TL	7	5	TL
rd100	9.0	8	4	8	4	TL	8	5	TL

Table C.3.: Results ATSPLIB with critical weights from $[\frac{\tau^*}{8}, \frac{\tau^*}{4}]$

				MTZ			SEC		
Instance name	Removed edges in %	UB (MCV heur)	LB (H_2-clique)	UB	LB	time (sec)	UB	LB	time (sec)
br17	83.8	6	6	6	6	1	6	6	1
ftv33	57.9	9	7	8	8	120	8	8	137
ftv35	53.3	9	6	8	8	205	8	8	149
ftv38	48.2	10	6	9	8	TL	9	9	2614
p43	25.0	4	2	4	3	TL	3	3	1398
ftv44	45.6	9	6	9	8	TL	9	9	3297
ftv47	40.6	11	5	11	8	TL	11	8	TL
ry48p	40.2	9	5	9	8	TL	9	8	TL
ft53	10.7	11	3	11	6	TL	11	6	TL
ftv55	40.3	10	5	10	7	TL	10	7	TL
ftv64	35.6	10	5	10	7	TL	10	7	TL
ft70	0.0	9	1	9	6	TL	9	5	TL
ftv70	29.3	11	5	11	6	TL	11	7	TL
kro124p	16.4	10	3	10	5	TL	10	5	TL

Table C.4.: Results ATSPLIB with critical weights from $[\frac{\tau^*}{6}, \frac{\tau^*}{2}]$

				MTZ			SEC		
Instance name	Removed edges in %	UB (MCV heur)	LB (H_2-clique)	UB	LB	time (sec)	UB	LB	time (sec)
br17	67.6	5	5	5	5	1	5	5	1
ftv33	23.9	7	4	7	6	TL	7	6	TL
ftv35	20.0	6	3	5	5	14852	6	5	TL
ftv38	11.7	7	3	7	5	TL	6	5	TL
p43	22.0	3	2	3	3	1	3	3	1
ftv44	11.6	8	3	8	5	TL	8	6	TL
ftv47	14.8	7	3	7	5	TL	7	6	TL
ry48p	14.0	7	3	7	5	TL	7	5	TL
ft53	0.6	7	2	7	4	TL	7	4	TL
ftv55	11.8	8	4	8	5	TL	8	5	TL
ftv64	10.0	8	3	8	5	TL	8	5	TL
ft70	0.0	7	1	7	4	TL	7	4	TL
ftv70	6.6	7	2	7	4	TL	7	5	TL
kro124p	0.6	7	2	7	4	TL	7	4	TL

Table C.5.: MCV for large TSPLIB instances with critical weights in $[\frac{\tau^*}{8}, \frac{\tau^*}{4}]$

Instance	MCV Basic		MCV Enhanced	
name	SolVal	time	SolVal	time(sec)
eil101	11	0.1	9	0.5
lin105	11	0.1	10	0.4
pr107	8	0.1	8	0.4
gr120	12	0.1	10	0.8
pr124	11	0.1	9	1.3
bier127	11	0.2	10	2.0
ch130	12	0.1	10	1.0
pr136	8	0.1	8	0.7
gr137	11	0.1	11	1.0
pr144	10	0.1	10	0.6
ch150	11	0.2	10	1.5
kroA150	10	0.1	10	1.2
kroB150	11	0.2	10	1.3
pr152	9	0.1	9	0.7
u159	11	0.2	10	1.3
si175	9	0.3	9	2.3
brg180	14	0.1	13	0.8
rat195	11	0.4	10	2.4
d198	9	0.6	9	3.6
kroA200	11	0.4	10	3.8
kroB200	10	0.4	10	3.8
gr202	11	0.7	10	9.9
ts225	11	0.5	11	3.3
tsp225	11	0.6	9	4.4
pr226	10	0.4	9	5.1
gr229	12	0.8	10	12.9
gil262	11	1.0	10	9.3
pr264	9	0.7	8	7.3
a280	11	1.1	10	7.9
pr299	11	1.2	10	12.0
lin318	11	1.9	10	13.0
linhp318	11	1.7	10	12.5
rd400	11	3.8	10	33.6
fl417	10	2.9	9	33.0
gr431	11	9.0	10	132.9
pr439	11	5.4	10	56.5
pcb442	11	5.3	10	34.5
d493	10	12.0	9	283.4
att532	11	11.3	10	69.4
ali535	11	13.2	10	431.2
si535	9	9.8	9	61.3
pa561	12	11.6	10	118.5
u574	11	12.3	9	99.8
rat575	11	10.9	10	73.1
p654	11	15.3	9	169.0
d657	12	17.3	10	205.2
gr666	11	24.8	10	220.2
u724	12	21.5	10	206.4
rat783	11	27.9	10	218.6
dsj1000	11	62.9	10	580.1

Table C.6.: MCV for large TSPLIB instances with critical weights in $[\frac{\tau^*}{6}, \frac{\tau^*}{2}]$

Instance	MCV Basic		MCV Enhanced	
name	SolVal	time	SolVal	time(sec)
eil101	8	0.1	7	0.7
lin105	9	0.1	7	1.0
pr107	6	0.1	6	0.5
gr120	8	0.2	7	1.2
pr124	8	0.1	8	2.2
bier127	8	0.3	7	2.4
ch130	7	0.2	7	1.4
pr136	8	0.2	7	1.4
gr137	9	0.2	7	2.2
pr144	8	0.2	8	1.5
ch150	8	0.3	7	2.3
kroA150	9	0.2	8	1.8
kroB150	8	0.2	7	1.8
pr152	7	0.2	7	1.7
u159	8	0.3	7	1.6
si175	7	0.5	6	0.9
brg180	14	0.1	13	0.8
rat195	9	0.6	7	3.4
d198	7	0.9	6	6.7
kroA200	8	0.6	7	4.7
kroB200	8	0.6	7	4.6
gr202	8	1.0	8	9.3
ts225	9	0.8	8	4.1
tsp225	8	0.9	7	5.6
pr226	9	0.7	8	8.9
gr229	8	1.2	7	7.0
gil262	8	1.3	7	10.2
pr264	8	1.3	6	11.9
a280	8	1.7	7	10.8
pr299	9	1.9	8	12.6
lin318	8	2.6	7	26.8
linhp318	8	2.5	7	22.7
logs	8	2.5	7	22.7
rd400	8	4.8	7	39.9
fl417	8	5.4	7	21.9
gr431	8	12.8	7	143.5
pr439	8	8.8	7	72.9
pcb442	8	7.2	7	51.2
d493	8	14.7	7	503.0
att532	8	15.0	7	156.5
ali535	9	21.8	7	277.8
si535	7	12.9	6	67.4
pa561	8	14.7	7	93.4
u574	9	14.0	7	134.8
rat575	8	14.3	8	114.9
p654	9	22.8	7	215.1
d657	9	24.5	7	211.8
gr666	8	34.3	7	625.4
u724	8	26.2	7	298.6
rat783	8	36.2	7	312.1
dsj1000	8	78.4	7	936.1

Table C.7.: MCV for large ATSPLIB instances with critical weights in $[\frac{\tau^*}{8}, \frac{\tau^*}{4}]$

Instance	MCV Basic		MCV Enhanced	
name	SolVal	time	SolVal	time(sec)
ftv170	14	0.2	13	2.1
rbg323	9	3.6	8	17.0
rbg358	11	5.9	10	29.0
rbg403	8	11.0	6	89.0
rbg443	8	16.8	7	84.7

Table C.8.: MCV for large ATSPLIB instances with critical weights in $[\frac{\tau^*}{6}, \frac{\tau^*}{2}]$

Instance	MCV Basic		MCV Enhanced	
name	SolVal	time	SolVal	time(sec)
ftv170	10	0.3	10	2.4
rbg323	7	4.3	6	24.7
rbg358	8	7.0	6	50.8
rbg403	6	13.3	5	115.0
rbg443	6	18.5	5	84.7

Bibliography

[1] Tobias Achterberg. *Constraint Integer Programming.* Doctoral thesis, Technische Universität Berlin, Fakultät II - Mathematik und Naturwissenschaften, Berlin, 2007.

[2] Peter Adam, Stefan Engelshove, Frank Heunemann, Thomas Thiemann, and Christoph von dem Busche. Hydrogen infrastructure - the pillar of energy transition. Technical report, Siemens Energy, Gascade Gastransport GmbH, Nowega GmbH, 2020. Accessed: 2021-09-17.

[3] Bernardetta Addis, Giuliana Carello, Andrea Grosso, and Elena Tànfani. Operating room scheduling and rescheduling: a rolling horizon approach. *Flexible Services and Manufacturing Journal*, 28(1):206–232, Jun 2016.

[4] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Upper Saddle River, NJ, USA, 1993.

[5] Ahmad B. Asghar, Stephen L. Smith, and Shreyas Sundaram. Multi-Robot Routing for Persistent Monitoring with Latency Constraints. In *2019 American Control Conference (ACC)*, pages 2620–2625, 2019.

[6] Charles Audet, Pierre Hansen, Brigitte Jaumard, and Gilles Savard. Links Between Linear Bilevel and Mixed 0-1 Programming Problems. *Journal of Optimization Theory and Applications*, 93(2):273–300, May 1997.

[7] Giorgio Ausiello, Marco Protasi, Alberto Marchetti-Spaccamela, Giorgio Gambosi, Pierluigi Crescenzi, and Viggo Kann. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties.* Springer, Berlin, Heidelberg, 1999.

[8] Jonathan F. Bard. *Practical Bilevel Optimization: Algorithms and Applications*, volume 30. Springer Science & Business Media, 1998.

[9] Jonathan F. Bard and James T. Moore. A Branch and Bound Algorithm for the Bilevel Programming Problem. *SIAM Journal on Scientific and Statistical Computing*, 11(2):281–292, 1990.

[10] Ahmad Baroutaji, Tabbi Wilberforce, Mohamad Ramadan, and Abdul Ghani Olabi. Comprehensive investigation on hydrogen and fuel cell technology in the aviation and aerospace sectors. *Renewable and Sustainable Energy Reviews*, 106:31–40, 2019.

[11] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958.

[12] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013.

[13] Piotr Berman and Toshihiro Fujito. On approximation properties of the Independent set problem for degree 3 graphs. In Selim G. Akl, Frank Dehne, Jörg-Rüdiger Sack, and Nicola Santoro, editors, *Algorithms and Data Structures*, pages 449–460, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[14] Timo Berthold. *Heuristic algorithms in global MINLP solvers.* Doctoral thesis, Technische Universität Berlin, Fakultät II - Mathematik und Naturwissenschaften, Berlin, 2015.

[15] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

[16] Tom Böttger, Veronika Grimm, Thomas Kleinert, and Martin Schmidt. The Cost of Decoupling Trade and Transport in the European Entry-Exit Gas Market. *European Journal of Operational Research*, 2021.

[17] Bundesnetzagentur and Bundeskartellamt. Monitoringbericht 2020. `https://www.bundesnetzagentur.de/SharedDocs/Mediathek/Berichte/2020/Monitoringbericht_Energie2020.pdf?__blob=publicationFile&v=7`, 2020. Accessed: 2021-09-17.

[18] Samuel Burer and Adam N. Letchford. Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97–106, 2012.

[19] Robert Burlacu, Herbert Egger, Martin Groß, Alexander Martin, Marc E. Pfetsch, Lars Schewe, Mathias Sirvent, and Martin Skutella. Maximizing the storage capacity of gas networks: A global MINLP approach. *Optimization and Engineering*, 20(2):543–573, June 2019.

[20] Robert Burlacu, Björn Geißler, and Lars Schewe. Solving mixed-integer nonlinear programmes using adaptively refined mixed-integer linear programmes. *Optimization Methods and Software*, 35(1):37–64, 2020.

[21] Herminia I. Calvete and Carmen Galé. *Algorithms for Linear Bilevel Optimization*, pages 293–312. Volume 1 of Dempe and Zemkoho [31], 2020.

[22] Frédéric Cazals and Chinmay Karande. A note on the problem of reporting maximal cliques. *Theoretical Computer Science*, 407(1):564–568, 2008.

[23] Günter Cerbe. *Grundlagen der Gastechnik. 7., vollständig neu bearbeitete Auflage*, volume 7. München, Wien, Hanser, 2008.

[24] Pablo Chamoso, William Raveane, Victor Parra, and Angélica González. UAVs Applied to the Counting and Monitoring of Animals. In Carlos Ramos, Paulo Novais, Céline Ehrwein Nihan, and Juan M. Corchado Rodríguez, editors, *Ambient Intelligence - Software and Applications*, pages 71–80, Cham, 2014. Springer International Publishing.

[25] Vašek Chvátal. *Linear Programming*. W. H. Freeman & Co Ltd, 1983.

[26] M Collins, L Cooper, R Helgason, J Kennington, and L LeBlanc. Solving the Pipe Network Analysis Problem Using Optimization Techniques. *Management Science*, 24(7):747–760, 1978.

[27] Stephen A. Cook. The Complexity of Theorem-Proving Procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, page 151–158, New York, NY, USA, 1971. Association for Computing Machinery.

[28] George Dantzig. Maximization of a linear function of variables subject to linear inequalities. *Activity Analysis of Production and Allocation*, 13:339–347, 1951.

[29] George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a Large-Scale Traveling-Salesman Problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954.

[30] George Dantzig and John Ramser. The Truck Dispatching Problem. *Management Science*, 6(1):80–91, 1959.

[31] Stephan Dempe and Alain Zemkoho. *Bilevel Optimization: Advances and Next Challenges*, volume 1. Springer, 2020.

[32] Martin Desrochers, Jan Karel Lenstra, Martin W. P. Savelbergh, and François Soumis. Vehicle Routing with Time Windows: Optimization and Approximation. *Vehicle Routing: Methods and Studies*, pages 65–84, 1988.

[33] Die Bundesregierung. Die Nationale Wasserstoffstrategie. Bundesministerium für Wirtschaft und Energie (BMWi), 2020. Accessed: 2021-09-2017.

[34] Edsger W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1:269–271, 1959.

[35] Paul E. Dodds and Stéphanie Demoullin. Conversion of the UK gas system to transport hydrogen. *International Journal of Hydrogen Energy*, 38(18):7189–7200, 2013.

[36] Pia Domschke, Bjorn Geißler, Oliver Kolb, Jens Lang, Alexander Martin, and Antonio Morsi. Combination of Nonlinear and Linear Optimization of Transient Gas Networks. *INFORMS Journal on Computing*, 23(4):605–617, 2011.

[37] Nir Drucker, Hsi-Ming Ho, Joël Ouaknine, Michal Penn, and Ofer Strichman. Cyclic-routing of Unmanned Aerial Vehicles. *Journal of Computer and System Sciences*, 103:18–45, 2019.

[38] Nir Drucker, Michael Penn, and Ofer Strichman. Cyclic routing of unmanned air vehicles. Technical report, Technion, Industrial Engineering and Management, IE/IS-2014-12, 2014.

[39] Nir Drucker, Michal Penn, and Ofer Strichman. Random instances for CR-UAV. `https://ie.technion.ac.il/~ofers/cruav/`. Accessed: 2021-09-17.

[40] Nir Drucker, Michal Penn, and Ofer Strichman. Cyclic Routing of Unmanned Aerial Vehicles. In Claude-Guy Quimper, editor, *Integration of AI and OR Techniques in Constraint Programming*, pages 125–141, Cham, 2016. Springer International Publishing.

[41] Jack Edmonds and Richard M. Karp. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *Journal of the ACM*, 19(2):248–264, April 1972.

[42] EU. Directive 2009/73/EC of the European Parliament and of the Council concerning common rules for the internal market in natural gas and repealing Directive 2003/55/EC. *OJ*, L 211:36–54, 2009.

[43] EU. Regulation No 715/2009 of the European Parliament and of the Council on conditions for access to the natural gas transmission networks and repealing Regulation No 1775/2005. *OJ*, L 211:94–136, 2009.

[44] Federal Ministry for Economic Affairs and Energy. Still indispensable for a reliable energy supply. `https://www.bmwi.de/Redaktion/EN/Dossier/conventional-energy-sources.html`, 2019. Accessed: 2021-09-17.

[45] Lester R. Ford junior. Network Flow Theory. Technical report, Rand Corp Santa Monica Ca, 1956.

[46] Forschungscampus - öffentlich-private Partnerschaft für Innovationen. `https://www.forschungscampus.bmbf.de/`. Accessed: 2021-09-17.

[47] José Fortuny-Amat and Bruce McCarl. A Representation and Economic Interpretation of a Two-Level Programming Problem. *Journal of the Operational Research Society*, 32(9):783–792, Sep 1981.

[48] Michael L. Fredman and Robert E. Tarjan. Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms. *J. ACM*, 34(3):596–615, July 1987.

[49] Armin Fügenschuh, Björn Geißler, Ralf Gollmer, Christine Hayn, René Henrion, Benjamin Hiller, Jesco Humpola, Thorsten Koch, Thomas Lehmann, Alexander Martin, Radoslava Mirkov, Antonio Morsi, Jessica Rövekamp, Lars Schewe, Martin Schmidt, Rüdiger Schultz, Robert Schwarz, Jonas Schweiger, Claudia Stangl, Marc C. Steinbach, and Bernhard M. Willert. Mathematical optimization for challenging network planning problems in unbundled liberalized gas markets. *Energy Systems*, 5(3):449–473, Sep 2014.

[50] Armin Fügenschuh, Björn Geißler, Ralf Gollmer, Antonio Morsi, Marc E. Pfetsch, Jessica Rövekamp, Martin Schmidt, Klaus Spreckelsen, and Marc C. Steinbach. Physical and Technical Fundamentals of Gas Networks. In Koch et al. [96].

[51] Armin Fügenschuh and Jesco Humpola. A Unified View on Relaxations for a Nonlinear Network Flow Problem. Technical Report 13-31, ZIB, Takustr.7, 14195 Berlin, 2013.

[52] Armin Fügenschuh, Konstanty Junosza-Szaniawski, and Sławomir Kwasiborski. The reservation-allocation network flow problem. Technical report, Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg, 2014.

[53] Internationale Gewässerschutzkommission für den Bodensee. Bodensee-Daten. `https://www.igkb.org/der-bodensee/seedaten/`. Accessed: 2021-08-02.

[54] Gerald Gamrath. *Enhanced predictions and structure exploitation in branch-and-bound*. Doctoral thesis, Technische Universität Berlin, Berlin, 2020.

[55] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1990.

[56] GasNZV 2010: Verordnung über den Zugang von Gasversorgungnetzen (Gasnetzzugangsverordnung - GasNZV). `http://bundesrecht.juris.de/bundesrecht/gasnzv_2010/gesamt.pdf`, 2010. Accessed: 2021-09-17.

[57] Björn Geißler. *Towards Globally Optimal Solutions for MINLPs by Discretization Techniques with Applications in Gas Network Optimization*. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg,, 07 2011.

[58] Björn Geißler, Alexander Martin, Antonio Morsi, and Lars Schewe. Using Piecewise Linear Functions for Solving MINLPs. In Jon Lee and Sven Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 287–314, New York, NY, 2012. Springer New York.

[59] Björn Geißler, Antonio Morsi, Lars Schewe, and Martin Schmidt. Solving power-constrained gas transportation problems using an MIP-based alternating direction method. *Computers & Chemical Engineering*, 82:303–317, 2015.

[60] Björn Geißler, Antonio Morsi, Lars Schewe, and Martin Schmidt. Solving Highly Detailed Gas Transport MINLPs: Block Separability and Penalty Alternating Direction Methods. *INFORMS Journal on Computing*, 30(2):309–323, 2018.

[61] Giorgio Giorgi and Tinne Hoff Kjeldsen. *Traces and Emergence of Nonlinear Programming*, volume 1. Springer Science & Business Media, 2013.

[62] Andrew V. Goldberg and Robert E. Tarjan. Finding minimum-cost circulations by canceling negative cycles. *Journal of the ACM*, 36(4):873–886, October 1989.

[63] Ángel M. González Rueda, Julio González Díaz, and María P. Fernández de Córdoba. A twist on SLP algorithms for NLP and MINLP problems: an application to gas transmission networks. *Optimization and Engineering*, 20(2):349–395, Jun 2019.

[64] Veronika Grimm, Lars Schewe, Martin Schmidt, and Gregor Zöttl. A multilevel model of the European entry-exit gas market. *Mathematical Methods of Operations Research*, 89(2):223–255, April 2019.

[65] Martin Gross, Marc E. Pfetsch, Lars Schewe, Martin Schmidt, and Martin Skutella. Algorithmic results for potential-based flows: Easy and hard cases. *Networks*, 73(3):306–324, 2019.

[66] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2. Springer Berlin Heidelberg, 1988.

[67] Martin Gugat, Günter Leugering, Alexander Martin, Martin Schmidt, Mathias Sirvent, and David Wintergerst. MIP-based instantaneous control of mixed-integer PDE-constrained gas transport problems. *Computational Optimization and Applications*, 70(1):267–294, May 2018.

[68] LLC Gurobi Optimization. Gurobi Optimizer Reference Manual, Version 9.0.0. http://www.gurobi.com, 2019.

[69] LLC Gurobi Optimization. Gurobi Optimizer Reference Manual, Version 9.1.2. http://www.gurobi.com, 2020.

[70] Dries Haeseldonckx and William D'haeseleer. The use of the natural-gas pipeline infrastructure for hydrogen transport in a changing market structure. *International Journal of Hydrogen Energy*, 32(10):1381–1386, 2007. EHEC2005.

[71] Pierre Hansen, Brigitte Jaumard, and Gilles Savard. New branch-and-bound rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing*, 13(5):1194–1217, 1992.

[72] Falk Hante, Günter Leugering, Alexander Martin, Lars Schewe, and Martin Schmidt. Challenges in Optimal Control Problems for Gas and Fluid Flow in Networks of Pipes and Canals: From Modeling to Industrial Applications. In Pammy Manchanda, René Lozi, and Abul Hasan

Siddiqi, editors, *Industrial Mathematics and Complex Systems: Emerging Mathematical Models, Methods and Algorithms*, Industrial and Applied Mathematics, pages 77–122. Springer Singapore, Singapore, 2017.

[73] Dieter Hausamann, Werner Zirnig, Gunter Schreier, and Peter Strobl. Monitoring of gas pipelines - a civil UAV application. *Aircraft Engineering and Aerospace Technology*, 77(5):352–360, 2005.

[74] Christine Hayn. *Computing maximal entry and exit capacities of transportation networks.* PhD thesis, PhD thesis. Friedrich-Alexander Universität Erlangen-Nürnberg, 2016.

[75] Kai Hennig and Robert Schwarz. Using Bilevel Optimization to find Severe Transport Situations in Gas Transmission Networks. Technical report, Zuse Institute Berlin, 2016.

[76] Felix Hennings. Benefits and Limitations of Simplified Transient Gas Flow Formulations. In Natalia Kliewer, Jan Fabian Ehmke, and Ralf Borndörfer, editors, *Operations Research Proceedings 2017*, pages 231–237, Cham, 2018. Springer International Publishing.

[77] Felix Hennings. Large-scale empirical study on the momentum equation's inertia term. *Journal of Natural Gas Science and Engineering*, page 104153, 2021.

[78] Felix Hennings, Lovis Anderson, Kai Hoppmann-Baum, Mark Turner, and Thorsten Koch. Controlling transient gas flow in real-world pipeline intersection areas. *Optimization and Engineering*, 22(2):687–734, Jun 2021.

[79] Felix Hennings, Milena Petkovic, and Tom Streubel. On the Numerical Treatment of Interlaced Target Values - Modeling, Optimization and Simulation of Regulating Valves in Gas Networks. Technical Report 21-32, ZIB, Takustr. 7, 14195 Berlin, 2021.

[80] Hsi-Ming Ho and Joël Ouaknine. The Cyclic-Routing UAV Problem is PSPACE-Complete. In Andrew Pitts, editor, *Foundations of Software Science and Computation Structures*, pages 328–342, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[81] Tom van der Hoeven. *Math in gas and the art of linearization.* PhD thesis, University of Groningen, 2004.

[82] Kai Hoppmann, Gioni Mexi, Oleg Burdakov, Carl Johan Casselgren, and Thorsten Koch. Minimum Cycle Partition with Length Requirements. In Emmanuel Hebrard and Nysret Musliu, editors, *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 273–282, Cham, 2020. Springer International Publishing.

[83] Kai Hoppmann and Robert Schwarz. Finding Maximum Minimum Cost Flows to Evaluate Gas Network Capacities. In Natalia Kliewer, Jan Fabian Ehmke, and Ralf Borndörfer, editors, *Operations Research Proceedings 2017*, pages 339–345, Cham, 2018. Springer International Publishing.

[84] Kai Hoppmann-Baum. On the Complexity of Computing Maximum and Minimum Min-Cost-Flows. *Networks*, 79(2):236–248, 2022.

[85] Kai Hoppmann-Baum, Oleg Burdakov, Gioni Mexi, Carl Johan Casselgren, and Thorsten Koch. Length-constrained cycle partition with an application to UAV routing*. *Optimization Methods and Software*, 0(0):1–37, 2022.

[86] Kai Hoppmann-Baum, Felix Hennings, Ralf Lenz, Uwe Gotzes, Nina Heinecke, Klaus Spreckelsen, and Thorsten Koch. Optimal Operation of Transient Gas Transport Networks. *Optimization and Engineering*, 22(2):735–781, Jun 2021.

[87] Kai Hoppmann-Baum, Felix Hennings, Janina Zittel, Uwe Gotzes, Eva-Maria Spreckelsen, Klaus Spreckelsen, and Thorsten Koch. From Natural Gas towards Hydrogen - A Feasibility Study on Current Transport Network Infrastructure and its Technical Control. Technical Report 20-27, ZIB, Takustr. 7, 14195 Berlin, 2020.

[88] Robert B. Jackson, Adrian Down, Nathan G. Phillips, Robert C. Ackley, Charles W. Cook, Desiree L. Plata, and Kaiguang Zhao. Natural Gas Pipeline Leaks Across Washington, DC. *Environmental Science & Technology*, 48(3):2051–2058, 2014.

[89] Izzet Karakurt, Gokhan Aydin, and Kerim Aydiner. Sources and mitigation of methane emissions by sectors: A critical review. *Renewable Energy*, 39(1):40–48, 2012.

[90] William Karush. *Minima of Functions of Several Variables with Inequalities as Side Conditions*, pages 217–245. Volume 1 of Giorgi and Kjeldsen [61], 2014.

[91] Leonid Genrikhovich Khachiyan. A polynomial algorithm in linear programming. In *Doklady Akademii Nauk*, volume 244, pages 1093–1096. Russian Academy of Sciences, 1979.

[92] Toh Chin Kiu, Thibankumar Arumugam, Saravanan Karuppanan, and Mark Ovinis. Failure pressure prediction of pipeline with single corrosion defect using artificial neural network. *Pipeline Science and Technology*, 4:10–17, 03 2020.

[93] Thomas Kleinert, Martine Labbé, Ivana Ljubić, and Martin Schmidt. A Survey on Mixed-Integer Programming Techniques in Bilevel Optimization. *EURO Journal on Computational Optimization*, page 100007, 2021.

[94] Thomas Kleinert, Martine Labbé, Fränk Plein, and Martin Schmidt. Technical Note - There's No Free Lunch: On the Hardness of Choosing a Correct Big-M in Bilevel Optimization. *Operations Research*, 68(6):1716–1721, 2020.

[95] Thomas Kleinert and Martin Schmidt. Why there is no need to use a big-M in linear bilevel optimization: A computational study of two ready-to-use approaches. submitted, 2020.

[96] Thorsten Koch, Benjamin Hiller, Marc E. Pfetsch, and Lars Schewe, editors. *Evaluating Gas Network Capacities*, volume 21 of *MOS-SIAM Series on Optimization*. SIAM, 2015.

[97] Oliver Kolb, Jens Lang, and Pia Bales. An implicit box scheme for subsonic compressible flow with dissipative source term. *Numerical Algorithms*, 53(2):293–307, March 2010.

[98] Bernhard Korte and Jens Vygen. *Combinatorial Optimization. Theory and applications*, volume 3. Springer Berlin Heidelberg, 2006.

[99] Harold W. Kuhn and Albert W. Tucker. *Nonlinear Programming*, pages 247–258. Volume 1 of Giorgi and Kjeldsen [61], 2014.

[100] Martine Labbé, Patrice Marcotte, and Gilles Savard. A Bilevel Model of Taxation and Its Application to Optimal Highway Pricing. *Management Science*, 44(12-part-1):1608–1622, 1998.

[101] Martine Labbé, Fränk Plein, and Martin Schmidt. Bookings in the European gas market: characterisation of feasibility and computational complexity results. *Optimization and Engineering*, 21(1):305–334, Mar 2020.

[102] Martine Labbé, Fränk Plein, Martin Schmidt, and Johannes Thürauf. Deciding Feasibility of a Booking in the European Gas Market on a Cycle is in P for the Case of Passive Networks. *Networks*, 78(2):128–152, 2021.

[103] Ailsa H. Land and Alison G. Doig. An Automatic Method of Solving Discrete Programming Problems. *Econometrica*, 28(3):497–520, 1960.

[104] Jon Lee and Sven Leyffer, editors. *Mixed Integer Nonlinear Programming*. Springer New York, 2012.

[105] Christian Liebchen and Rolf H. Möhring. The Modeling Power of the Periodic Event Scheduling Problem: Railway Timetables – and Beyond. In Frank Geraets, Leo Kroon, Anita Schoebel, Dorothea Wagner, and Christos D. Zaroliagis, editors, *Algorithmic Methods for Railway Optimization*, pages 3–40, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[106] Kimberley Lindenberg. Comparing severe gas transport situations through the network: Similarity or reduction methods. Master's thesis, Delft University of Technology, 2015.

[107] June Liu, Yuxin Fan, Zhong Chen, and Yue Zheng. *Methods for Pessimistic Bilevel Optimization*, pages 403–420. Volume 1 of Dempe and Zemkoho [31], 2020.

[108] Thomas L. Magnanti and Richard T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1984.

[109] Debora Mahlke, Alexander Martin, and Susanne Moritz. A simulated annealing algorithm for transient optimization in gas networks. *Mathematical Methods of Operations Research*, 66:99 – 116, 2007.

[110] Terrence W. K. Mak, Pascal Van Hentenryck, Anatoly Zlotnik, Hassan Hijazi, and Russell Bent. Efficient dynamic compressor optimization in natural gas transmission systems. In *2016 American Control Conference (ACC)*, pages 7484–7491, 2016.

[111] A. Migdalas, Panos M. Pardalos, and Peter Värbrand. *Multilevel Optimization: Algorithms and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2012.

[112] Clair E. Miller, Albert W. Tucker, and Richard A. Zemlin. Integer Programming Formulation of Traveling Salesman Problems. *Journal of the ACM*, 7(4):326–329, October 1960.

[113] Forschungscampus MODAL. `http://forschungscampus-modal.de/`. Accessed: 2021-09-17.

[114] Susanne Moritz. *A Mixed Integer Approach for the Transient Case of Gas Network Optimization*. PhD thesis, Technische Universität, Darmstadt, Februar 2007.

[115] Deutschland Erdgas: Verbrauch 1965 - 2020. `https://www.ceicdata.com/de/indicator/germany/natural-gas-consumption`. Accessed: 2021-09-17.

[116] Lars K. Nielsen, Leo Kroon, and Gábor Maróti. A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, 220(2):496–509, 2012.

[117] Philipp Niemann, Laura Bittner, Philipp Schrögel, and Christiane Hauser. Science Slams as Edutainment: A Reception Study. *Media and Communication*, 8(1):177–190, 2020.

[118] Johann Nikuradse. *Laws of Flow in Rough Pipes*. National Advisory Committee for Aeronautics Washington, 1950.

[119] Fred M. Odom and Gordon L. Muster. Tutorial On Modeling of Gas Turbine Driven Centrifugal Compressors. In *PSIG Annual Meeting*, 05 2009.

[120] Open Grid Europe GmbH. `https://oge.net/`. Accessed: 2021-09-17.

[121] Open Grid Europe GmbH. `https://oge.net/en/us/company/our-history`. Accessed: 2021-09-17.

[122] Anibal Ollero, Jose Ramiro Martínez de Dios, and Luis Merino. Unmanned aerial vehicles as tools for forest-fire fighting. *Forest Ecology and Management*, 234:S263, 2006.

[123] Andrej J. Osiadacz. Different Transient Flow Models - Limitations, Advantages, And Disadvantages. In *PSIG Annual Meeting*, 10 1996.

[124] Alexander Otto, Martin Robinius, Thomas Grube, Sebastian Schiebahn, Aaron Praktiknjo, and Detlef Stolten. Power-to-Steel: Reducing CO_2 through the Integration of Renewable Energy and Hydrogen into the German Steel Industry. *Energies*, 10(4), 2017.

[125] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[126] Pápay, J. A Termeléstechnológiai Paraméterek Változása a Gáztelepek Muvelése Során. *OGIL Musz. Tud. Kozl.*, pages 267–273, 1968.

[127] Jaap Pedersen, Kai Hoppmann-Baum, Janina Zittel, and Thorsten Koch. Blending hydrogen into natural gas: An assessment of the capacity of the German gas grid; Technical Report. Technical Report 21-21, ZIB, Takustr. 7, 14195 Berlin, 2021.

[128] Ivan Penn. The Next Energy Battle: Renewables vs. Natural Gas. `https://www.nytimes.com/2020/07/06/business/energy-environment/renewable-energy-natural-gas.html`. Accessed: 2021-09-17.

[129] Milena Petkovic, Ying Chen, Inken Gamrath, Uwe Gotzes, Natalia Selini Hadjidimitrou, Janina Zittel, Xiaofei Xu, and Thorsten Koch. A hybrid approach for high precision prediction of gas flows. *Energy Systems*, Aug 2021.

[130] Milena Petkovic, Thorsten Koch, and Janina Zittel. Deep learning for spatio-temporal supply and demand forecasting in natural gas transmission networks. Technical Report 21-01, ZIB, Takustr. 7, 14195 Berlin, 2021.

[131] Marc E. Pfetsch. *The maximum feasible subsystem problem and vertex-facet incidences of polyhedra*. Doctoral thesis, Technische Universität Berlin, Fakultät II - Mathematik und Naturwissenschaften, Berlin, 2003.

[132] Salvador Pineda and Juan Miguel Morales. Solving Linear Bilevel Problems Using Big-Ms: Not All That Glitters Is Gold. *IEEE Transactions on Power Systems*, 34(3):2469–2471, 2019.

[133] Fränk Plein, Johannes Thürauf, Martine Labbé, and Martin Schmidt. A bilevel optimization approach to decide the feasibility of bookings in the European gas market. *Mathematical Methods of Operations Research*, Sep 2021.

[134] Carlo Raucci. *The potential of hydrogen to fuel international shipping*. PhD thesis, UCL (University College London), 2017.

[135] Gerhard Reinelt. TSPLIB and ATSPLIB. http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/. Accessed: 2021-09-17.

[136] Gerhard Reinelt. TSPLIB - A Traveling Salesman Problem Library. *ORSA Journal on Computing*, 3(4):376–384, 1991.

[137] Markus Reuß, Thomas Grube, Martin Robinius, and Detlef Stolten. A hydrogen supply chain with spatial resolution: Comparative analysis of infrastructure technologies in Germany. *Applied Energy*, 247:438–453, 2019.

[138] R. Tyrrell Rockafellar. *Network Flows and Monotropic Optimization*. Athena Scientific, 1998.

[139] Daniel J. Rosenkrantz, Richard E. Stearns, and Philip M. II Lewis. An Analysis of Several Heuristics for the Traveling Salesman Problem. *SIAM Journal on Computing*, 6(3):563–581, 1977.

[140] Jessica Rövekamp. Background on gas market regulation. In Koch et al. [96].

[141] Roger Z. Ríos-Mercado and Conrado Borraz-Sánchez. Optimization problems in natural gas transportation systems: A state-of-the-art review. *Applied Energy*, 147:536–555, 2015.

[142] Nikolaos V. Sahinidis. Mixed-integer nonlinear programming 2018. *Optimization and Engineering*, 20(2):301–306, Jun 2019.

[143] Sartaj Sahni and Teofilo Gonzalez. P-Complete Approximation Problems. *J. ACM*, 23(3):555–565, July 1976.

[144] Marcella Samà, Andrea D'Ariano, and Dario Pacciarelli. Rolling Horizon Approach for Aircraft Scheduling in the Terminal Control Area of Busy Airports. *Procedia - Social and Behavioral Sciences*, 80:531–552, 2013. 20th International Symposium on Transportation and Traffic Theory (ISTTT 2013).

[145] Hendrik Scharf, Fabian Arnold, and Dominic Lencz. Future natural gas consumption in the context of decarbonization - A meta-analysis of scenarios modeling the German energy system. *Energy Strategy Reviews*, 33:100591, 2021.

[146] Lars Schewe, Martin Schmidt, and Johannes Thürauf. Computing technical capacities in the European entry-exit gas market is NP-hard. *Annals of Operations Research*, 295(1):337–362, Dec 2020.

[147] Martin Schmidt, Denis Aßmann, Robert Burlacu, Jesco Humpola, Imke Joormann, Nikolaos Kanelakis, Thorsten Koch, Djamal Oucherif, Marc E. Pfetsch, Lars Schewe, Robert Schwarz, and Mathias Sirvent. GasLib – A Library of Gas Network Instances. *Data*, 2(4):article 40, 2017.

[148] scienceslam.de. `https://www.scienceslam.de/`. Accessed: 2021-08-04.

[149] Paolo Serafini and Walter Ukovich. A Mathematical Model for Periodic Scheduling Problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581, 1989.

[150] J. Cole Smith and Churlzu Lim. *Algorithms for Network Interdiction and Fortification Games*, pages 609–644. Springer New York, New York, NY, 2008.

[151] J. Cole Smith and Yongjia Song. A survey of network interdiction models and algorithms. *European Journal of Operational Research*, 283(3):797–811, 2020.

[152] Marius M. Solomon and Jacques Desrosiers. Survey Paper - Time Window Constrained Routing and Scheduling Problems. *Transportation Science*, 22(1):1–13, 1988.

[153] Jarig J. Steringa, Marco Hoogwerf, and Harry Dijkhuis. A Systematic Approach to Transmission Stress Tests in Entry-Exit Systems. In *PSIG Annual Meeting*, 05 2015.

[154] Maria Stimm. *Science slam: Ein Format der Wissenschaftskommunikation aus erwachsenenpädagogischer Perspektive*, volume 2. transcript Verlag, 2019.

[155] Detlef Stolten and Bernd Emonts. *Hydrogen Science and Engineering, 2 Volume Set: Materials, Processes, Systems, and Technology*, volume 1. John Wiley & Sons, 2016.

[156] John W. Suurballe and Robert E. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14(2):325–336, 1984.

[157] Alberto Varone and Michele Ferrari. Power to liquid and power to gas: An option for the German Energiewende. *Renewable and Sustainable Energy Reviews*, 45:207–218, 2015.

[158] Heinrich Von Stackelberg. *Marktform und Gleichgewicht*. Springer, 1934.

[159] Anthony Wang, Kees van der Leun, Daan Peters, and Maud Buseman. European hydrogen backbone: How a dedicated hydrogen infrastructure can be created, 2020.

[160] WELT. Explosion an Gasstation in Österreich - ein Toter, viele Verletzte. `https://www.welt.de/vermischtes/article171502283/Explosion-an-Gasstation-in-Oesterreich-ein-Toter-viele-Verletzte.html`, 2017. Accessed: 2021-09-17.

[161] Bernhard Willert. *Validation of nominations in gas networks and properties of technical capacities*. PhD thesis, Hannover: Gottfried Wilhelm Leibniz Universität Hannover, 2014.

[162] Sonja Wogrin, Salvador Pineda, and Diego A. Tejada-Arango. *Applications of Bilevel Optimization in Energy and Electricity Markets*, pages 139–168. Volume 1 of Dempe and Zemkoho [31], 2020.

[163] R. Kevin Wood. Bilevel Network Interdiction Models: Formulations and Solutions. *Wiley Encyclopedia of Operations Research and Management Science*, 2011.

[164] Christopher Yang and Joan Ogden. Determining the lowest-cost hydrogen delivery mode. *International Journal of Hydrogen Energy*, 32(2):268–286, 2007.

[165] Wei Yu, Zhaohui Liu, and Xiaoguang Bao. New approximation algorithms for the minimum cycle cover problem. *Theoretical Computer Science*, 793:44–58, 2019.

[166] ZEIT ONLINE. Russland stoppt Gaslieferungen an die Ukraine. `https://www.zeit.de/politik/ausland/2015-11/gasstreit-russland-gas-ukraine-stop-lieferungen`, 2015. Accessed: 2019-09-17.

[167] Anatoly Zlotnik, Michael Chertkov, and Scott Backhaus. Optimal Control of Transient Flow in Natural Gas Networks. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 4563–4570, 2015.