Mixed Integer Models for the Optimisation of Gas Networks in the Stationary Case

Vom Fachbereich Mathematik

der Technischen Universität Darmstadt

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

genehmigte Dissertation

von

Dipl.–Math. Markus Möller

aus Rotenburg a. d. Fulda

Referent:	Prof. Dr. A. Martin
Korreferent:	Prof. Dr. G. Leugering
Tag der Einreichung:	01.03.2004
Tag der mündlichen Prüfung:	14.05.2004

Darmstadt 2004
D17

Etwas erkennen nach dem was es ganz an und für sich sei, ist für alle Ewigkeit unmöglich: weil es sich widerspricht. Denn sobald ich erkenne, habe ich eine Vorstellung: diese muß aber eben weil sie meine Vorstellung ist verschieden seyn von dem Erkannten und kann nicht mit demselben identisch seyn.

Im Reiche der Wirklichkeit, so schön, glücklich und anmuthig sie auch ausgefallen seyn mag, bewegen wir uns doch stets nur unter dem Einfluß der Schwere . . . hingegen sind wir, im Reiche der Gedanken, unkörperliche Geister, ohne Schwere und ohne Noth.

Schopenhauer

Abstract

Through out the world the natural gas resources will be one of the most important sources of energy in the future. The development of optimised possibilities for the distribution of gas through a network of pipelines will be very important for an effective operation of a gas transmission network. The aim of this thesis is to formulate this problem as a suitable mathematical mixed integer problem and to find advanced solutions, using techniques of mixed integer programming.
The main problem of the so called Transient Technical Optimisation (TTO) is to minimise the total supply costs of a gas transmission company that has to satisfy demands of different kinds. A gas network basically consists of a number of compressors and valves that are connected via pipes. The gas transmission companies dispatchers decide how to run the compressors and how to switch the valves cost-efficiently such that all demands of all customers are satisfied.
The cost function mainly consists of the supply costs of driving the compressors. Note that the compressors consum a fraction of the gas transported through the pipelines. The costs imposed by consumed gas should be minimised.
The gas transmission network has to satisfy several demands that are described by a minimal or maximal pressure requirement at a certain node or in a pipe. Also the consumers want to get gas of a certain volume and quality. Furthermore some physical constraints, like Kirchhoff's laws have to be modelled. There are also some combinatorial constraints, e.g. the different possibilities of switching the valves or compressor configurations. Note, that some of the constraints are nonlinear, like the pressure loss in a pipeline or the fuel-gas consumption of the compressors. In order to formulate TTO as a mixed integer program we approximate the nonlinear constraints by piecewise linear functions.
Considering the experiences of other projects where mixed integer programs have been used, e.g. VLSI-Design or Telecommunications, we know that the problem can be solved by examination of the underlying polyhedra of such complex and high-dimensional mixed integer programs. We know from earlier test evaluations of smaller problems that it is not possible to solve real gas transmission problems with state-of-the-art general mixed integer programming solvers. One programming approach is the search of better valid (or even facet-defining) inequalities of the polyhedra for the use in a Branch-and-Cut Algorithm. We have developed a new class of valid inequalities that have been integrated in a general MIP solver algorithm.

Summarising the results it was possible to develop a polynomial separation algorithm for a special class of polyhedra. The use of these cuts reduces the calculation time by a significant factor. A suitable branch-and-bound algorithm is also added. The cuts and the branching algorithms have been tested on several test-models of real gas-networks.

Zusammenfassung

Die reichen Gasvorkommen der Erde werden in den nächsten Jahren eine der Hauptenergiequellen unserer Gesellschaft darstellen. Aus diesem Grund gewinnt die Suche nach optimierten Verfahren, große Gasmengen effizient durch weitverzweigte Gasnetze transportieren zu können immer größere Bedeutung. Das Ziel der vorliegenden Arbeit, dieses Problem, das als Transiente Technische Optimierung (TTO) bezeichnet wird, in Form eines gemischt ganzzahligen linearen Optimierungsproblems so zu formulieren, dass die Kosten, die einem Gasversorgungsunternehmen bei der Gasverteilung in einem Gasnetz entstehen, minimiert werden.

Das Hauptproblem der Transienten Technischen Optimierung (TTO) besteht darin, die Gesamtheit der Verteilungskosten eines Gasversorgungsunternehmens zu minimieren, so dass alle Anforderungen, die an das Gasnetz gestellt sind, erfüllt werden können. Ein Gasnetzwerk besteht im Wesentlichen aus einer Menge von Kompressoren (Kompressorstationen) und Ventilen, die über Leitungen miteinander verbunden sind. Die Kompressoren werden dazu benutzt, um den in den Gasleitungen durch Reibung entstehenden Druckabfall wieder auszugleichen. Die Dispatcher der Gasversorgungsunternehmen müssen Entscheidungen darüber treffen, wie die Kompressoren und die Ventile kosteneffizient geschaltet werden können, so dass alle Bedingungen, die aufgrund physikalischer oder marktgegebener Situationen an das Gasnetz gestellt werden, erfüllt werden können.

Die Kostenfunktion besteht in der Hauptsache aus der Summe der Betriebskosten der einzelnen Verdichter. Hierbei ist zu bedenken, dass die Kompressoren einen gewissen Anteil des durch die Gasleitungen transportierten Gases verbrauchen. Die Kosten und damit die Menge des benötigten Gases sollen minimiert werden.

Das Gasnetzwerk muss zusätzlich einige weitere Bedingungen erfüllen, die beispielsweise darin bestehen, dass in einem Knoten oder in einer Leitung ein minimaler Druck nicht unterschritten und ein maximaler Druck nicht überschritten werden darf. Desgleichen ergeben sich aus der Strömungsmechanik physikalische Bedingungen, die ein Gasnetz erfüllen muss, ebenso gelten Erhaltungsgleichungen, wie sie durch die Kirchhoffschen Gesetze beschrieben werden. Besondere Bedeutung bei der Formulierung des Problems der Transienten Technischen Optimierung als gemischt ganzzahliges Optimierungsproblem haben die auftretenden kombinatorischen Nebenbedingungen, z.B. die verschiedenen Möglichkeiten, die Ventile zu schalten oder die verschiedenen Fahrwege von Kompressoren. Hierbei besteht eine wichtige Problematik darin, dass einige Bedingungen nichtlinear sind, wie z.B. der Druckverlust innerhalb der einzelnen Leitungen oder der Brenngasverbrauch der Kompressoren. Um das Problem der TTO als ein gemischt ganzzahliges Programm formulieren zu können, approximieren wir die nichtlinearen Nebenbedingungen durch stückweise lineare Funktionen.

Wenn wir die Erfahrungen, die in anderen Projekten, bei denen gemischt ganzzahlige Modelle zur Modellierung eines Problems genutzt wurden, heranziehen (so z.B. im VLSI-Design oder in der Telekommunikation), so war zu erwarten, dass auch das Problem der TTO schnell und effizient gelöst werden kann, wenn die Polyeder der zugrundeliegenden komplexen und hochdimensionalen gemischt ganzzahligen Probleme analysiert werden. Denn Erfahrungen mit früheren Testrechnungen anhand kleiner und stark vereinfachter Gasnetze zeigten, dass es unmöglich ist, reale Gasnetzoptimierungsprobleme mit derzeitig üblichen Standardlösern für allgemeine gemischt ganzzahlige Programme zu lösen. Der Ansatz, der daher in dieser Arbeit verfolgt wird, besteht in der Suche nach besseren gültigen (evtl. sogar facettendefinierenden) Ungleichungen der zugrundeliegenden (Teil-) Polyeder als Voraussetzung

für die Anwendung von LP-basierten Branch-and-Bound Verfahren, wobei die LP-Relaxierung durch Schnittebenen verschärft wird (sog. Branch-and-Cut oder Schnittebenenverfahren). Die in dieser Arbeit entwickelten Typen von gültigen Ungleichungen wurden in einen allgemeinen Löser für gemischt ganzzahlige Modelle integriert.

Insgesamt konnte ein polynomialer Separierungsalgorithmus für eine spezielle Klasse von Polyedern entwickelt werden. Die Anwendung dieser Schnitte kann die Rechenzeit deutlich reduzieren. Ein ebenfalls entwickeltes Branch-and-Bound Verfahren vervollständigt das erarbeitete Schnittebenenverfahren. Die Schnitte und die Branchingalgorithmen wurden anhand der Berechnungen bei einigen kleineren Gasnetzen getestet.

Acknowledgements

To start with I would like to express my deep gratefulness to all who gave me their support while I was working on this thesis.

Especially I want to express my great gratitude to Prof. Dr. A. Martin who is the person without whose help I would have never been able to complete the work on this thesis. I would also like to thank the BMBF (the German ministry for education and scientific research) which gave the financial support for the work on project 03MAM5DA. This thesis is a consequence of the research work of this project. Without the support of the BMBF this thesis would not have been written.

My special thanks go to the members of the Discrete Optimisation Research Group of the Technical University Darmstadt, especially to Dipl. Math. Susanne Moritz for the discussions and cooperation during the work on the problem.

Furthermore I would like to thank my co-referee Prof. Dr. G. Leugering and Prof. Dr. J. Lang for the appropriation for his tool KARDOS which was used during the implementation of the algorithms that were developed for the problem and the contact persons of the Ruhrgas AG and the PSI AG for their expert support in the task of gas networks, especially Dipl. Math. K. Reith and Dr. E. Sekirnjak.

Darmstadt, February 2004 M.Möller

Contents

Chapter 1

Introduction

Natural gas is a mixture of different hydrocarbons. It consists about 90% of methane (CH_4) which is the simplest alkane to be found in nature. Methane is found under the earth's crust but it also arises in processes of fermentation under the absence of air, e.g., in marshes or in purification plants. Also there are some non inflammable substances in natural gas such as carbon dioxide (CO_2), helium (He) and nitrogen (N_2). Before the gas can be transported to the consumers it has to be cleaned from pollutions (for instance water (H_2O), sulfur (S) or hydrogen sulfide (H_2S)). Significant natural gas sources are found all over the world - the biggest ones in F.R. Russia. The typical use of natural gas is in household (heating, cooking , ...), in industry (production of heat, electricity, cooling, ...), in provisions of services and in road traffic. Since natural gas is ecologically compatible it has become to a very important energy source in Germany over the last years [9], [2]. In 1965 only 1% of the German energy consumption has been satisfied by the use of natural gas. Today this rate increased up to 20% and the gas consumption still will increase rapidly over the next years [42], [44].
Another very important fact is the forthcoming liberalisation of the european gas-market in the closer future. On the long term only those companies which are able to react to the demands and requirements of the global market will survive. Because of this it is necessary to develop control systems that are able of compiling and editing the data of a gas network. Also simulation and optimisation tools are of great importance.
The summary of all these facts was the principal reason for our attempt to search for suitable models for the optimisation of gas networks.

1.1 The Problem

A gas network basically consists of a set of compressors and valves that are connected by pipes. The task of the Transient Technical Optimisation is to optimise the drives of the gas and control the compressors cost-efficiently in such a manner that the required demands are satisfied. This problem leads to a complex mixed integer nonlinear optimisation problem. We approach it by approximating the nonlinearities by piecewise linear functions leading to a huge mixed integer program. We study the polyhedral consequences of this model and present some new cutting planes. Our computational results show the benefits when incorporating these cuts into a general mixed integer programming solver.

Let us describe the problem a bit more detailed: The situation is that the pressure of flowing gas decreases in pipes due to the friction with the pipes walls. The consumers want to get gas of a certain pressure and volume and with a certain quality. So it is necessary to unwind the pressure loss in the pipes. This is usually done by using compressors. The problem is that the compressors consume some fraction of the gas flowing through the pipes. Our task is to develop a suitable mathematical model for this situation and we want to optimise the drives of the gas and run the compressors cost-efficiently such that all demands are satisfied. In order to show how complex this problem is we consider the dimensions

of the gas network which is driven by the German Ruhrgas AG. The approximate length of the pipes is about 11000 kilometres and there are 26 compressor stations each of this consisting of several singular compressors [2]. The number of additional valves or control valves in this network is immense. Until now it is not possible to optimise this complex gas network using a mathematical optimisation tool. All optimisation nowadays has to be done completely by the gas company's dispatchers. Additionally the problem must be calculated in a very short time since the situation in the gas network very often changes.
Posing the problem we have to consider the different facets of the model: There are a lot of nonlinear aspects. The most important consist in the hydraulic of the pipes (the already mentioned pressure loss) and the fuel gas consumption of the compressors. So the parameters of the compressors are nonlinear functions, e.g. the efficiency or the specific heat rate. Also the gas quality, the compound of gas and gas delivering contracts lead to nonlinear functions. There are also combinatorial aspects. Valves and compressors can be switched on or off. In compressor stations one can decide in which combinations several compressors should be run. Also delivery contracts can lead to combinatorial aspects. Of course the problem is time dependent in fact as the hydraulics of pipes is transient. Another important component of the model is that usually stochastical aspects have to be considered. So the sales quantity of gas depends on the time of day or on the season; but the weather is uncertain and for the control of the gas network it is of course very important if a winter is cold or open. Also some of the model-parameters are uncertain. A very important problem in the last years was the planned liberalisation of the european gas-market which also would have lead to a complex stochastical situation. But, until now the liberalisation of gas market was by far not as wide as the liberalisation of electricity market [42].

We cannot consider all important aspects we have mentioned above because the problem would have become unsolvable. In the present work we could only regard the most important parts of the problem and we tried a special mathematical approach by mixed integer linear programming for the stationary case of the problem. We only wanted to show that our approach can be useful in order to solve a very complicated physical and technical problem.
In Chapter 2 we describe the physical and in Chapter 3 the mathematical basics of our problem. In the Chapter 4 we formulate a suitable mixed integer problem. Later, in Chapter 5 we describe what can be considered as main part of this thesis: the polyhedral consequences of the linearisation of the nonlinear functions of the model and we develop a new separation algorithm. Some consequences and extensions of these algorithms are studied in Chapter 6. In the last chapter we deal with some implementation details and give some computational results.

1.2 Previous Approaches to the Problem

Additionally we want to give a short outline about the previous approaches which have been attempted in order to attack the problem of the Transient Technical Optimisation. As far as we know, until today no algorithm has been developed that includes all nonlinear, combinatorial and stochastical aspects of the problem; only parts of the problem have been considered.
Often dynamic programming has been used [14],[51] but the problem is that the gas network must have an easy structure in this case. Ideal for this approach is a gun-barrel system, i.e., a directed graph consisting of pipes and compressors without any cycles. Recent papers on this topic try to deal with the possibility of cyclic gas networks for the stationary case [36].
Since the problem contains nonlinear aspects (e.g. the pressure loss in pipes) nonlinear optimisation methods are frequently applied. Here the nonlinear functions are modelled correctly neglecting the combinatorial aspects of the problem. This nonlinear optimisation problems are often solved by subgradient techniques [19],[24],[45] or SQP-methods [11].
Sometimes well known heuristics like Simulated Annealing or genetic algorithms are used, see [36], [50].
Let us give a more detailed description of approaches that use similar techniques as in this thesis is done.

The approach presented in this thesis is related to an approach for the stationary case presented in [35]. Here the authors describe a computer program for the stationary optimisation they have developed. The model the authors use is also based on a huge mixed integer program. The optimisation problem is solved by an iterative method. The nonlinear gas flow equations are linearised in order to build a linear problem. The problem is solved iteratively until convergence is obtained. The cost function e.g. includes the fuel gas consumption of the compressors or the gas flow from the sources. The solution is computed in two steps: In the first step the linear problem is solved by mixed integer linear programming. After that the algorithm re-linearises around the calculated optimum until convergence is obtained. The authors give several computed examples. In [43] the problem is examined by developing an extended simplex algorithm. Here also mixed integer linear programming is used (the nonlinear functions are linearised and then this model is iterated). In [38], [39] the method of sequential optimisation is used which also leads to a mixed integer problem. We remark that these papers gave us important hints for our own activities since we were lucky to work together with its author at the beginning of our work on the gas optimisation problem. Also here a mixed integer linear problem for the gas optimisation problem is developed. We remark that this model is quite different from the model presented here. Especially the approximation of nonlinear functions is quite different. The author uses the technique of sequential optimisation. Here in every iteration a solution of the linearised model (use of linear Taylor approximations) is calculated. For every variable the author defines a slack variable in order to ensure the obtainment of a solution for the problem. Every slack variable is part of the cost function of the model and a penalty factor should ensure that the optimisation process converges to the problem's optimum.
Another mixed integer formulation of the problem is presented in [16]. Here the author does not consider a mixed integer model for a short time optimisation but for medium- and long-term optimisation as is pointed out in the calculation of some examples. The author also uses the concept of SOS type 2 inequalities but this is done quite differently from the way this concept is used in this thesis.
Unfortunately none of the approaches models all important facts of the problem - this of course also holds for the ideas presented in this thesis.
Nevertheless, the approach exposed in this thesis is to the best of our knowledge new in the sense that we are not aware of any IP- or MILP-approaches of the TTO-Problem which use the techniques of polyhedral studies combined with branch-and-cut algorithms. Also our use of the SOS type 2 inequalities seems to be new. Added to this nobody seems to have used the present branching strategies. The advantage of our model is that we do not use an iterative optimisation process and we can guarantee not to terminate with only a local optimum which is a danger of all nonlinear optimisation techniques and the other described iterative methods. Also our linearisation methods for nonlinear functions can be used very generally and are not dependent of the considered problem. We further add that the techniques and concepts we described often have been helpful in order to solve complex mixed integer programs (see e.g. [30],[29], [23], [12], [15]) and so there was hope also for the problem of the Transient Technical Optimisation.

We remark that this thesis was written in a cooperative scientific project of the Technical University Darmstadt (here the formulation and analysis of mixed integer programs for TTO was done as described in this thesis), the Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB) where new methods of Nonlinear Optimisation were developed and the University of Duisburg where the problem of TTO was studied in consideration of stochastic phenomena ([46]).

Chapter 2

Physical Background

2.1 Introduction

In this chapter we give a short outline about the physical and technical preliminaries that are necessary for understanding the mixed integer model which we developed for the problem of the Transient Technical Optimisation. From physics we know that the gas transport in a network is described as a system of partial differential equations. At first we define the variables and constants that are needed in order to describe this PDE-system. Later on we will see that the most depending variables must become constants in order to give us the possibility to linearise the system. All simplifications we will do later on are precise enough for the requirements of the gas industry. Furthermore we want to derive some important simplification formulas in this chapter especially the pressure loss formula. Let us describe now the needed variables. The basic facts of the following descriptions are developed and extended from [39],[40], [41].

2.2 Summary of Variables

To describe the TTO we must introduce variables that depend on time and space. We will denote the space variable by x, the time variables are indicated with t.

x	variable of (pipe) length	$[m]$
t	variable of time	$[s]$

After that we carry on with the summary of the dynamic variables. The most important variables we will use in our model are the gas flow density (flow-rate, gas volume flow) q, the gas pressure p, the power N and the fuel gas consumption f of a compressor.

$q = q(x,t)$	gas flow density/flow-rate in a pipe	$[m^3/s]$
$v = v(x,t)$	gas velocity	$[m/s]$
$p = p(x,t)$	pressure of gas	$[Pa]$
$\rho = \rho(x,t)$	gas density	$[kg/m^3]$
$h = h(x)$	height of pipe	$[m]$
$\lambda = \lambda(q)$	value of pipe friction	[1]
$W = W(t)$	heat addition to gas	$[kJ/(kg\, s)]$
$A = A(x)$	slice plane of pipe	$[m^2]$
$T = T(x,t)$	gas temperature	$[K]$
$z = z(p,T)$	compressibility factor (z-factor)	[1]
$p_r = p_r(x,t)$	relatively gas pressure	[1]
$T_r = T_r(x,t)$	relatively gas temperature	[1]
$M = M(x,t)$	mass flow of gas	$[kg/s]$

$R_e = R_e(x,t)$	Reynolds number of gas	$[1]$
p_{in}	gas pressure at the beginning of pipe/compressor	$[bar]$
p_{out}	gas pressure at the end of pipe/compressor	$[bar]$
T_{in}	gas temperature at the beginning of pipe/compressor	$[K]$
z_{in}	z-factor at the beginning of pipe/compressor	$[1]$
H_{ad}	adiabatic head of a compressor	$[m]$
N_{th}	theoretical power of a compressor	$[kW]$
N	power of a compressor	$[kW]$
η_{ad}	adiabatic efficiency of a compressor	$[1]$
f	fuel gas consumption of a compressor	$[nm^3/h]$

After the depending variables we have to consider general physical constants and also typical gas constants which are shown in the following table.

g	acceleration due to gravity	$[m/s^2]$
L	length of pipe	$[m]$
D	diameter of pipe	$[m]$
c_v	specific heat	$[kJ/(kg\,K)]$
ρ_0	norm density	$[kg/m^3]$
z_0	norm compressibility coefficient	$[1]$
p_0	norm gas pressure	$[bar]$
T_0	norm temperature	$[K]$
p_c	pseudo-critical pressure	$[bar]$
T_c	pseudo-critical temperature	$[K]$
k	pipe roughness	$[m]$
s	barometric factor	$[1]$
κ	adiabatic coefficient	$[1]$
m	molecular weight of gas	$[kg/kmol]$
R	universal gas constant	$[kJ/(kmol\,K)]$
b	specific heat rate	$[1]$
H_u	calorific value of gas	$[kW/m^3]$
η	dynamic viscosity of gas	$[kg/(m\,s)]$

The following three equations describe the physics of the gas in a pipe: the continuity equation, the momentum equation and the energy equation. Let us give some basic facts on these equations.

2.3 The Continuity Equation

The first important PDE (Partial Differential Equation) is the continuity equation: The alteration of the gas flow is commensurate with the alteration of the gas density.

$$A\,\frac{\partial \rho}{\partial t} + \frac{\partial M}{\partial x} = 0$$

We see that the continuity equation has only influence if we consider the transient case. In the stationary case the continuity equation reduces to

$$\frac{\partial M}{\partial x} = 0 \qquad \Rightarrow \qquad M = const.$$

Since M can be converted in q (a technical variable usually used in gas networks) by application of the gas density via the formula

$$M = \rho_0\; q$$

we also get

$$q = const.$$

We will take q as gas flow variable in our model.

2.4 The Momentum Equation

The momentum equation describes all forces that operate on the gas molecules. Here we show the momentum equation in the form for cylindrical pipes:

$$\frac{\partial p}{\partial x} + g\rho \frac{\partial h}{\partial x} + \frac{\lambda |v| v}{2D}\rho + \frac{1}{A}\frac{\partial M}{\partial t} + \frac{\partial(\rho v^2)}{\partial x} = 0.$$

The first term in this equation describes the pressure in dependence of the pipe length, the second term describes the gravitational force working on the gas molecules for inclined pipes. After that the friction force follows and the fourth term in the equation describes the change of flow rate in time. The last term considers the so called impact pressure.

2.5 The Energy Equation

The third PDE is the energy equation:

$$W \cdot (\rho A\, dx) = \frac{\partial}{\partial t}[(\rho A\, dx) \cdot (c_v T + \frac{v^2}{2} + g \cdot dh)]$$

$$+ \frac{\partial}{\partial x}[(\rho v A\, dx) \cdot (c_v T + \frac{p}{\rho} + \frac{v^2}{2} + g \cdot dh)].$$

W describes the heat addition (per mass flow and time) from the soil to the gas and c_v the specific heat per constant gas volume.
The energy equation deals with the connection of the inner energy of the gas and the heat exchange with the soil.
The most important components of the energy equation are (see [39]): The gas very slowly emits energy to the soil. Furthermore the energy equation describes the kinetic energy which describes fast activities of the gas. Also the Joule Thomson Effect that describes highly inclinations of the pipes is part of the energy equation.

From these facts we conclude: Since in Germany the gas pipelines are placed subterranean and the variation in temperature is very small this PDE can serendipitously be neglected. But even without this PDE the problem is difficult enough.

2.6 The Gas Equation

The gas equation is the fourth basic equation we have to deal with. We begin with the gas equation which models the ideal gas:

$$\frac{\rho}{\rho_0} = \frac{p}{T}\frac{z_0 T_0}{p_0}.$$

Since we want to deal with the behaviour of real gas we have to modify this formula. Several basic approaches have been developed in order to model real gas, e.g. the Van-der-Waals gas equation.
We use another approach which is used by the gas engineers: we introduce the so called z-factor (compressibility factor) which is a nonlinear function depending on gas pressure and temperature. The z-factor describes the differences of a real gas and the ideal gas. In our model we will simplify the

z-factor. So we get as a constitutive equation for a real gas:

$$\frac{\rho}{\rho_0} = \frac{p}{z(p,T)T}\frac{z_0 T_0}{p_0}.$$

Clearly we get a ideal gas if $z(p,T) \equiv 1$.
The American Gas Association (AGA) has developed a formula which is a good approximation for the z-factor for a gas pressure up to $70\ bar$. This value is usually not exceeded in our gas network. In order to give this empirical formula we first define the relatively gas pressure p_r

$$p_r = \frac{p}{p_c}$$

and analogously the relatively gas temperature T_r

$$T_r = \frac{T}{T_c}$$

and so we get the following formula:

$$z(p,T) = 1 + 0.257\,p_r - 0.533\frac{p_r}{T_r}.$$

2.7 Simplification of the Momentum Equation

In our model we have to deal with the two important nonlinearities of the pressure loss in pipes and the fuel gas consumption of a compressor. Here we want to conclude a formula for the pressure loss in pipes. This formula is a consequence of the momentum equation (here for cylindrical pipes) that we already know:

$$\frac{\partial p}{\partial x} + g\rho\frac{\partial h}{\partial x} + \frac{\lambda|v|v}{2D}\rho + \frac{1}{A}\frac{\partial M}{\partial t} + \frac{\partial(\rho v^2)}{\partial x} = 0.$$

In the stationary case we get $\frac{\partial M}{\partial t} = 0$. For horizontal pipes also the effect of the impact pressure and the gravity can be neglected (see [39], [10]) and the formula simplifies to:

$$\frac{\partial p}{\partial x} = -\lambda\frac{|v|v}{2D}\rho. \tag{2.1}$$

With the equation

$$v = \frac{\rho_0 q}{A\rho}$$

we can write this equation as

$$\frac{\partial p}{\partial x} = -\lambda\frac{\rho_0^2}{A^2}\frac{|q|q}{2D}\frac{1}{\rho}. \tag{2.2}$$

Now we have to strike out a bit:
First we define the Reynolds number R_e that describes if the gas flow is laminar or turbulent. It holds

$$R_e = \frac{4}{\pi\,\eta}\frac{M}{D}.$$

Remember that η means the dynamic viscosity of the gas. Generally the gas flow in the gas networks we are considering is turbulent.
The pipe friction value λ is a very important part of the right-hand side of equation (2.2). It is only possible to give an implicit formula for λ:

$$\frac{1}{\sqrt{\lambda}} = -2log_{10}(\frac{2.51}{R_e\sqrt{\lambda}} + \frac{k}{3.71\,D}).$$

The solution can be calculated by a suitable iteration method. We will show on page 18 that for usual values in gas networks this method will converge.

But let us now come back to our actual aim:
We can write the gas equation in the form

$$\rho = \frac{\rho_0 \, p}{z(p,T)T} \frac{z_0 T_0}{p_0}$$

and together with the average values

$$\begin{aligned}
q_m &= \frac{q_{out} + q_{in}}{2}, \\
T_m &= \frac{T_{out} + T_{in}}{2}, \\
z_m &= z(p_m, T_m), \\
\frac{\rho_m}{\rho_0} &= \frac{p_m}{z_m T_m} \frac{z_0 T_0}{p_0}, \\
\lambda_m &= \lambda(q_m).
\end{aligned}$$

we can solve (2.2). We remark that the calculation of the average pressure p_m in the pipe is a little bit more difficult and so we tried several formulas which are used in the modelling of gas networks. We will talk about this later. We also mention that in our calculations we usually set $q_{out} = q_{in} = q$ since we assume the gas flow q to be constant in the pipe. So we do not need to introduce gas flow variables in the beginning- or endpoints of a segment.
Now we conclude from (2.2)

$$\frac{z_0 T_0}{z_m \, p_0} \int_{p_{in}}^{p_{out}} p \, dp = -\frac{\lambda_m \rho_0 |q_m| q_m T_m}{2DA^2} \int_0^L 1 \, dx$$

and from this we can calculate the following simplified pressure drop calculation:

$$p_{out}^2 = p_{in}^2 - \frac{\lambda_m L}{DA^2} \frac{\rho_0 p_0 z_m T_m}{z_0 T_0} |q_m| q_m \,, \tag{2.3}$$

where subindex m indicates the average numbers defined above.
The same formula can also be derived by a discretisation of $\frac{\partial p}{\partial x}$ (take the difference quotient instead of the differential quotient) which is pointed out in [10].

Defining the value ff as

$$\text{ff} = \frac{\lambda_m L}{DA^2} \frac{\rho_0 p_0 z_m T_m}{z_0 T_0}$$

and setting the gas flow q in the pipe to q_m (which is of course a very crude approximation) we can write approximation formula (2.3) as

$$p_{out}^2 = p_{in}^2 - \text{ff} \, q \, |q| \tag{2.4}$$

and in the case (we will usually consider) that in the pipe there is no reversion of the gas flow direction we get

$$p_{out}^2 = p_{in}^2 - \text{ff} \, q^2. \tag{2.5}$$

In our stationary model we assume q_m and so λ_m to be constant. We also do with constant temperature and so T_m is constant. Our pressure loss formula simplifies if we also assume z_m to be constant because in this situation ff becomes constant. Indeed, for first test calculations this can be done as we have seen in our test networks. Now we can examine the pressure loss in a serial connection or in a parallel connection of pipes when using formula (2.5). We conclude the following marginal

Lemma 1 *Consider a serial connection of* $n_s > 1$ *pipes with* constant *values* $f\!f_i$ *for* $i = 1, \ldots, n_s$. *For every pipe we approximate the gas pressure at the end of this pipe by formula (2.5). Is the gas pressure at the end of the last pipe physically reasonable than we can substitute these pipes by* a single *pipe with value*

$$f\!f = \sum_{i=1}^{n_s} f\!f_i$$

and the gas pressure at the end of this pipe is equal to the gas pressure at the end of the last pipe of the serial connection.

Proof. The proof is easily done by induction because we can show that the pressure p_{out,n_s} after the last pipe of the serial connection is

$$p_{out,n_s} = \sqrt{p_{in}^2 - (\sum_{i=1}^{n_s} \mathrm{ff}_i)q^2}$$

which is equal to the gas pressure at the end of the pipe with value ff. □

For a parallel connection of some pipes holds

Lemma 2 *Consider a parallel connection of* $n_p > 1$ *pipes with* constant *values* $f\!f_i$ *and gas flow* q_i *for* $i = 1, \ldots, n_p$. *For every pipe we again approximate the gas pressure at the end of this pipe by formula (2.5). Is the gas pressure at the end of the pipes physically reasonable (and so of course equal) which implies*

$$f\!f_i q_i^2 = f\!f_j q_j^2 \qquad \forall i, j = 1, \ldots, n_p$$

than we can substitue these pipes (depending on the gas flow) by a single *pipe with value*

$$f\!f = \frac{f\!f_1 q_1^2}{(\sum_{i=1}^{n_p} q_i)^2}$$

and the gas pressure at the end of this pipe is the same as at the end of the pipes of the parallel connection.

Proof. The proof is done by an easy calculation. □

As a short forecast we add that the analysis of polyhedron $P_\triangle$ defined at the beginning of Chapter 5 is important and cannot be reduced by the lemma presented here. Although we gave a formula in order to replace a serial connection of pipes by a single one it will be useful to develop separation algorithms also for $P_\triangle$. Also remember that these formulas only hold for constant ff! Often the pressure values at the connecting nodes of two pipes are important for the opimisation and the cuts that result of this polyhedron will be helpful for the optimisation of a complexer model. We also mention that serial connections of a huge number of pipes usually do not exist in real gas networks. Another advantage of the analysis of $P_\triangle$ is that in this polyhedron we are not dependent on using approximation formulas. In $P_\triangle$ we can use exact solutions of the Momentum Equation. Also $P_\triangle$ can be generalised to complexer situations as we will describe in Chapter 5.

There are several possibilities to calculate the average pressure p_m which is needed for the calculation of z_m. In the gas industry often the following formula is used (for a calculation see [40]):

$$p_m = \frac{2}{3}(p_{in} + p_{out} - \frac{p_{in}p_{out}}{p_{in} + p_{out}}). \tag{2.6}$$

Taking the simplified pressure drop formula we can conclude the following algebraic equation of degree 3 in order to calculate the gas pressure p_{out} at the end of a pipe:

$$p_{out}^3 + (p_{in} + c_2)p_{out}^2 + (c_2 p_{in} + c_1 - p_{in}^2)p_{in} - p_{in}^3 + c_2 p_{in}^2 + c_1 p_{in} = 0$$

with

$$c_1 = (\frac{4}{\pi})^2 \frac{L\rho_0 T_m p_0}{D^5 z_0 T_0} \lambda_m \, q_m^2$$

and

$$c_2 = \frac{2}{3} c_1 (\frac{0.257}{p_c} - \frac{0.533 T_c}{p_c T_m}).$$

Here we combined formula (2.6) and formula (2.3).
We have implemented the last formula in our model and calculated the solution by the formula of Cardano.
It is well known that the approximation

$$p_m = \frac{1}{2}(p_{in} + p_{out}) \tag{2.7}$$

is more inexact than the approximation (2.6) but when we are using this formula in our simplified pressure drop calculation the values for p_{out} (in our test models) are quite the same. Since the calculation of p_{out} becomes easier in this way we have decided to use (2.7).

We remark that it is also possible to derive a formula like (2.2) in the case of a inclined pipe. Let s describe the altitude difference between the beginning of the pipe and the end of the pipe. We get

$$p_{out}^2 = (p_{in}^2 - \frac{\lambda_m L}{DA^2} \frac{\rho_0 p_0 z_m T_m}{z_0 T_0} |q_m| q_m \, \frac{e^s - 1}{s}) e^{-s}.$$

Because of $\lim_{s \to 0} \frac{e^s - 1}{se^s} = 1$ for $s = 0$ formula (2.3) also follows from the last formula.

Let us shortly give a reason why the iteration method for calculating λ usually will converge: We can write the implicit formula for λ as:

$$\lambda = \frac{1}{4(log_{10}(\frac{2.51}{R_e \sqrt{\lambda}} + \frac{k}{3.71 D}))^2}.$$

Define a function $g(\lambda) = \lambda$ and $h(\lambda) = \frac{1}{4(log_{10}(\frac{2.51}{R_e\sqrt{\lambda}} + \frac{k}{3.71D}))^2}$. Then we can construct an iteration method as described in [5], pp.744. It is well known that this method converges if $|h'(\lambda)| < 1$. We get

$$h'(\lambda) = \frac{2.51}{4 \, ln(10) R_e \lambda^{\frac{3}{2}} (\frac{2.51}{R_e\sqrt{\lambda}} + \frac{k}{3.71\,D}) \, (log_{10}(\frac{2.51}{R_e\sqrt{\lambda}} + \frac{k}{3.71\,D}))^3}.$$

In our situations we can neglect $\frac{k}{3.71\,D}$ and from practical experiences in gas networks we can assume $R_e \geq 10000$ and $\lambda \geq 100$. Under these conditions we can calculate

$$|h'(\lambda)| \approx |\frac{1}{4\,ln(10)\,\lambda\,(log_{10}(\frac{2.51}{R_e\sqrt{\lambda}}))^3}| < 1$$

since the logarithm functions are strict monotone.

2.8 The Behaviour of Compressors

2.8.1 The ideal Compressor

In this section we want to discuss the technical facts of an ideal compressor because we want to minimise the fuel gas consumption of the compressor.

Starting from the adiabatic height H_{ad} of a compressor which can be interpreted as the height on which the gas had to be pumped up for the same power

$$H_{ad} = \frac{\kappa}{\kappa - 1} \frac{z_{in} \, R T_{in}}{g \, m} ((\frac{p_{out}}{p_{in}})^{\frac{\kappa-1}{\kappa}} - 1)$$

the formula of the fuel consumption f is given by

$$f = \frac{N \, b}{1000 \, H_u}$$

with

$$\begin{aligned} N_{th} &= \frac{g}{3600} \, H_{ad} \, \rho_0 \, q, \\ N &= \frac{N_{th}}{\eta_{ad}}. \end{aligned}$$

So we can write the formula for the fuel gas consumption f in the following form:

$$f = \gamma((\frac{p_{out}}{p_{in}})^{\frac{\kappa-1}{\kappa}} - 1)q \tag{2.8}$$

with

$$\gamma = \frac{\rho_0 b}{3.6 \cdot 10^6 \eta_{ad} H_u} \frac{\kappa}{\kappa - 1} \frac{z_{in} \, R T_{in}}{m}. \tag{2.9}$$

We remark that the function which describes the fuel gas consumption of a compressor is neither concave nor convex which is interesting to know for building up our model because the sum of the fuel gas consumption of the compressors is the objective function of our model. In order to see this we formulate the following little

Lemma 3 *The function*

$$f = f(x, y, z) = \alpha((\frac{x}{y})^a + \beta)z$$

with $0 < a < 1, \alpha, x, y, z > 0, \beta \in \mathbb{R}$ *is neither concave nor convex.*

Proof. Define two functions $f_1 : \mathbb{R}_+ \to \mathbb{R}$ with

$$f_1 = f_1(x) = \gamma(\alpha_1 x^a + \beta)$$

with $\gamma, \alpha_1 > 0$ and $f_2 : \mathbb{R}_+ \to \mathbb{R}$ with

$$f_2 = f_2(x) = \gamma(\alpha_2 x^{-a} + \beta)$$

with $\gamma, \alpha_1 > 0$. We get:

$$f''_1(x) = \gamma\alpha_1 a(a - 1)x^{a-2} < 0$$

and

$$f''_2(x) = \gamma\alpha_2 a(a + 1)x^{-a-2} > 0.$$

So f_1 is concave and f_2 is convex. Since f_1 and f_2 can be understood obviously as restrictions of the function f cannot be concave or convex. □

Since the value of the adiabatic exponent κ fulfils $0 < \frac{\kappa-1}{\kappa} < 1$ the lemma above shows us that the fuel gas consumption is neither a convex nor concave function. We have to mention this because the piecewise linearisation of such functions as constraints of a Mixed Integer Program has to be done very carefully [20]. We also mention that function f does not have relative minima or maxima.

As a short addendum we also for compressors examine serial and parallel connections. Since in the formula of γ in (2.9) only η_{ad} and b are constants depending on the compressor we define for a compressor $\zeta = \frac{b}{\eta_{ad}}$ and now can write formula (2.8) as

$$f = \zeta\bar{\gamma}((\frac{p_{out}}{p_{in}})^{\frac{\kappa-1}{\kappa}} - 1)q \tag{2.10}$$

with

$$\bar{\gamma} = \frac{\rho_0}{3.6 \cdot 10^6 H_u} \frac{\kappa}{\kappa - 1} \frac{z_{in} R T_{in}}{m}. \tag{2.11}$$

Then we conclude the following

Lemma 4 *Consider a parallel connection of n_p compressors with values ζ_i and gas flow q_i for $i = 1, \ldots, n_p$. For every compressor we calculate the fuel gas consumption by formula (2.10). Then we theoretically can substitute these compressors by* a single *compressor with value*

$$\zeta = \frac{\sum_{i=1}^{n_p} \zeta_i q_i}{\sum_{i=1}^{n_p} q_i}$$

and the fuel gas consumption of this compressor is the same as the sum of the fuel gas consumption of all compressors of the parallel connection.

Proof. The proof is done by an easy calculation. □

It is easy to see that in the case of a compressor no easy formula for a (theoretical) simplification of a serial connection can be given.

2.8.2 The general Compressor

In contrast to the ideal compressor we have to consider that the adiabatic efficiency η_{ad} and the specific heat rate b of the compressor is not constant. It holds $\eta_{ad} = \eta_{ad}(Q, H_{ad})$ and $b = b(n, N, T_L)$. Here T_L is the temperature of air that is sucked in by the compressor. Also the maximal (N_{max}) and minimal (N_{min}) power of the compressor are not constant values. The maximal power is a function depending on the revolution number $n = n(Q, H_{ad}, T_L)$ of the compressor. Also the increase of the gas temperature in the compressor has to be considered in a real compressor.
We remark that the technical value Q is defined as

$$Q = \frac{1}{3.6} \frac{p_0 T_{in} z_{in}}{p_{in} T_0} q.$$

So the situation of a real compressor is much more complicated as in the case of an ideal compressor.

Usually the data of a real compressor cannot be calculated via explicit formulas. The data are stored in so called characteristic diagrams which are used from the gas transmission companies [39].

The first characteristic diagram usually has the following shape:

$$\begin{array}{ccccccccc}
 & & n_1 & & n_2 & \cdots & \cdots & & n_j \\
\eta_1 & H_{ad1,1} & Q_{1,1} & H_{ad1,2} & Q_{1,2} & \cdots & \cdots & H_{ad1,j} & Q_{1,j} \\
\eta_2 & H_{ad2,1} & Q_{2,1} & H_{ad2,2} & Q_{2,2} & \cdots & \cdots & H_{ad2,j} & Q_{2,j} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\eta_i & H_{adi,1} & Q_{i,1} & H_{adi,2} & Q_{i,2} & \cdots & \cdots & H_{adi,j} & Q_{i,j}
\end{array}$$

Here for every pair of a efficiency and a revolution value of the compressor values for the adiabatic head and the gas flow are given. Via the formula for the adiabatic head H_{ad} the connection to the pressure

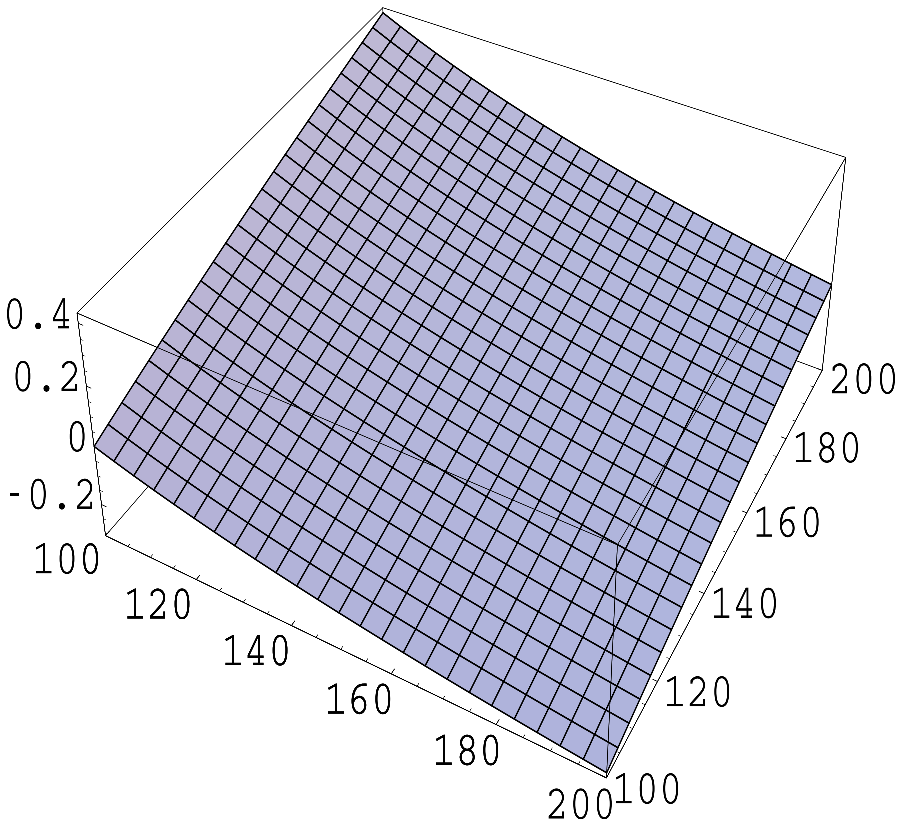

Figure 2.1: Plot of the function f in Lemma 3 with fixed variable z

values p_{in} and p_{out} is given.
The second characteristic diagram usually has the following shape:

$$\begin{array}{cccccc} & N_{th,1} & N_{th,2} & \dots & N_{th,j} \\ & f_1 & f_2 & \dots & f_j \\ & n_1 & n_2 & \dots & n_j \\ T_1 & N_{1,1} & N_{1,2} & \dots & N_{1,j} \\ T_2 & N_{2,1} & N_{2,2} & \dots & N_{2,j} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ T_i & N_{i,1} & N_{i,2} & \dots & N_{i,j} \end{array}$$

This diagram is to be understood in the following way: For discrete shaft power values (we mean N_{th}) of the compressor the values of fuel gas consumption and the revolution number of the compressor is given (above part of the diagram). Now for every pair of a shaft power value and a certain temperature of the sucked in air the maximal power of the compressor is given in the second part of the table.
In this thesis we work with an ideal compressor. For a general compressor usually only the first characteristic diagram is used for calculations. For the implementation of the model and the mathematical analysis it is enough to discuss the ideal compressor.

A good collection of basic facts of gas dynamics can be found in [52].

Chapter 3

Mathematical Background

3.1 Introduction

We want to formulate the problem of the optimisation of gas networks with methods of discrete optimisation especially in form of a Mixed Integer Linear Problem (MILP). Because of this we now want to give a short summary of the basic mathematical concepts which we have used in the present work. It is our aim to be so detailed that it is possible to understand our researches even for these readers who did not work before with Mixed Integer Programs. A deeper study of the topics can be achieved from pertinent text books on Optimisation, i.e., [7], [8], [22], [33], [34], [37], [49], [32], [47] and so on. We follow [31] for our short recapitulation of the basic facts.

3.2 Problems in Discrete Optimisation

At first we want to define some classes of optimisation problems. We will see that a Mixed Integer Linear Program is a special subclass of the General Optimisation Problem which we define now.

Definition 5 *General Optimisation Problem.*
Let S be a set and $(T, \leq)$ an ordered set, i.e., for all $s, t \in T$ holds $s < t$, $s > t$ or $s = t$. Further let $f : S \to T$ be a map. We are searching for an element $x^* \in S$ with $f(x^*) \geq f(x)$ $\forall x \in S$ (Maximisation Problem) or for an element $x^* \in S$ with $f(x^*) \leq f(x)$ $\forall x \in S$ (Minimisation Problem). Shortly we write

$$\max_{x \in S} f(x) \qquad \textit{resp.} \qquad \min_{x \in S} f(x) \tag{3.1}$$

For $S = \emptyset$ the values above are not defined.

Typical examples for T are $\mathbb{R}$, $\mathbb{Q}$ or $\mathbb{Z}$.

We give some examples for S:

(a) Let $g_i\,,\ i \in \{1, 2, \ldots, m\}$ and $h_j\,,\ j \in \{1, 2, \ldots, p\}$ be continuous (differentiable) functions from $\mathbb{R}^n$ in $\mathbb{R}$. Then we call (3.1) with

$$S \;=\; \Big\{ x \in \mathbb{R}^n \;\big|\; g_i(x) \leq 0,\ i = 1, \ldots, m;\ h_j(x) = 0,\ j = 1, \ldots, p \Big\}$$

a `Nonlinear Optimisation problem`.

(b) Let f be convex, i.e.,

$$\lambda f(x) \;+\; (1 - \lambda) f(y) \;\geq\; f(\lambda x \;+\; (1 - \lambda) y) \quad \forall x, y \in S\,,\ 0 \leq \lambda \leq 1.$$

Is furthermore S also convex, i.e.,

$$\lambda x \; + \; (1-\lambda)y \; \in \; S \quad \forall x, y \in S, \; 0 \leq \lambda \leq 1$$

(e.g., S can be given by the following set $S = \{x|\, g_i(x) \, \leq \, 0\}$, where g_i are convex functions), we call (3.1) a `convex Optimisation problem`.

Now we come to the problem class we need to formulate our gas network optimisation problem. We define the

Definition 6 *Mixed Integer Linear problem.*
Let $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $p \in \{0, \dots, n\}$.
Then we call

$$\begin{aligned} \min \; c^T x & \\ Ax \leq b & \\ x \in \mathbb{Z}^p \; \times \mathbb{R}^{n-p} & \end{aligned} \tag{3.1}$$

`Mixed Integer Linear Program (MILP, MIP)`. *In the case* $p = 0$ *the MIP is called* linear program (LP) *and in the case* $p = n$ *we call it* integer program (IP).

3.3 General Notation

In the next sections we are working with the following well known basic sets of numbers: $\mathbb{N} = \{1, 2, \dots\}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, that means the sets of natural, integer, rational and real numbers.
Let us give some notation on vectors and matrices:
Let $\mathbb{R}^n$ be the set of n-tuples (vectors) with components from $\mathbb{R}$.
We understand vectors always in the form of a column

$$x \;=\; \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \;\in\; \mathbb{R}^n \;.$$

For a row vector we write x^T. A vector $x \in \mathbb{R}^n$ is greater than or equal 0 ($x \geq 0$) iff all of its components are greater or equal 0. A vector $x \in \mathbb{R}^n$ is greater than 0 ($x > 0$) iff all of its components are greater 0. We define the scalar product as $x^T\, y \;=\; \sum\limits_{i=1}^{n} x_i\, y_i$ and the euclidic norm by $||x|| \;=\; \sqrt{x^T\, x}$.
Furthermore we introduce the following notations for the canonical basis vectors and the vector $\mathbb{1}$ (with value 1 for every component (we will specialise these definitions later on)):

$$\begin{aligned} e_j \;&=\; \text{j-th canonical basis vector,} \\ \mathbb{1} \;&=\; \sum_{j=1}^{n} e_j \;=\; (1, 1, \dots, 1)^T, \end{aligned}$$

and for matrices we denote:

$$\mathbb{R}^{m\times n} \;\;:\;\; \text{set of all } m \times n \text{ matrices with entries from } \mathbb{R}.$$

For $A \in \mathbb{R}^{m\times n}$ we also write

$$\begin{aligned} A \;&=\; (a_{ij})_{\substack{\text{i=1,...,m} \\ \text{j=1,...,n}}}\,, \\ A_{\cdot j} \;&=\; \begin{pmatrix} a_{1j} \\ \vdots \\ a_{mj} \end{pmatrix} \;:\; \text{j-th column of } A, \\ A_{i\cdot} \;&=\; (a_{i1}, \dots, a_{in}) \;:\; \text{i-th row of } A. \end{aligned}$$

For two sets $I \subseteq \{1,\dots,m\} = M,\ J \subseteq \{1,\dots,n\} = N$ we denote by

$$A_{IJ} = (a_{ij})_{\substack{i \in I \\ j \in J}}$$

the sub-matrix of $A \in \mathbb{R}^{m \times n}$, which consists of the rows of set I and the columns of set J. The matrices $A_{.J}$ and $A_{I.}$ are given by $A_{.J} = A_{MJ}$ resp. $A_{I.} = A_{IN}$. For a vector $b \in \mathbb{R}^n$ we define analogously $b_i = b_{i.}$ and $b_I = b_{I.}$.

Combinations of Vectors, Hulls and Independency

A vector $x \in \mathbb{R}^n$ is called `linear combination` of the vectors $x_1, x_2, \dots, x_k \in \mathbb{R}^n$, if there exists a vector $\lambda \in \mathbb{R}^k$ with

$$x = \sum_{i=1}^{k} \lambda_i x_i.$$

If additionally holds

$$\lambda \geq 0 \text{ and } \lambda^T \mathbb{1} = 1 \quad \text{, we call } x \text{ convex combination.}$$

The combination is called `pure` if $\lambda \neq 0$ and $\lambda \neq e_j \, \forall j \in \{1,\dots,k\}$. For a subset $\emptyset \neq S \subseteq \mathbb{R}^n$ we call

$$\left.\begin{array}{l}\text{lin(S)}\\ \text{conv(S)}\end{array}\right\} \text{ the } \left\{\begin{array}{l}\text{linear}\\ \text{convex}\end{array}\right\} \text{ hull of } S,$$

i.e., the set of all vectors, which can be written as linear (convex) combination of a finite number of vectors of S. We define

$$\begin{aligned}\text{lin}(\emptyset) &= \{0\},\\ \text{conv}(\emptyset) &= \emptyset.\end{aligned}$$

A subset $S \in \mathbb{R}^n$ is called

$$\left\{\begin{array}{l}\text{linear space (vector space)}\\ \text{convex set}\end{array}\right\} \text{, if } S = \left\{\begin{array}{l}\text{lin(S)}\\ \text{conv(S)}\end{array}\right\}.$$

A subset $\emptyset \neq S \subseteq \mathbb{R}^n$ is called `linear independent` if no element from S can be written as a pure linear combination of elements from S. The empty set is not linear independent. Every set which is not linear dependent is called `linear independent`. For $S \subseteq \mathbb{R}^n$ we call the `cardinality` of the biggest linear independent subset of S the rang of S. We denote the rang by $rang(S)$. The rang of a matrix A is defined as the rang of the set of column vectors, which is equal the rang of the set of row vectors which is well known from linear algebra. For the rang of matrix A we shortly write $rg(A)$. For $A \in \mathbb{R}^{m \times n}$ holds $rg(A) \leq \min\{m, n\}$. If even holds $rg(A) = \min\{m, n\}$ we say that A has `full rang`.

3.4 Graph Notations

As usual we define a graph as a pair $D = (V, E)$. V is a nonempty finite set and E a set of pairs of V, i.e., it holds $E \subseteq \{(u, v) | u, v \in V, u \neq v\}$. We call the elements of V vertices and the elements of E edges of D.

We assume D to be a directed graph, i.e., we did not define $E \subseteq \{\{u, v\} | u, v \in V, u \neq v\}$. The reason

for this is that in our model the gas flow is directed in every segment.
For all $v \in V$ we denote with $\delta^-(v)$ the set of all outgoing edges in v. Formally we write

$$\delta^-(v) = \{e \in E \mid \exists u \in V : e = (v,u)\}.$$

Analogously we denote for all $v \in V$ with $\delta^+(v)$ the set of all ingoing edges in v. Formally we write

$$\delta^+(v) = \{e \in E \mid \exists u \in V : e = (u,v)\}.$$

3.5 Basics of the Theory of Polyhedra

Now we define some basic topics which are necessary in order to understand the techniques to tighten the formulation of a Mixed-Integer Linear Problem. Very important is to define a polyhedron and a polytop. Closed to these concepts are the definition of hyper- and half-spaces since every polyhedron can be understood as intersection of finite many half-spaces. Also hyper- and half-spaces are special polyhedra.

Definition 7 *Hyperspace, Halfspace, Polyhedron, Polytop.*

(a) A subset $G \subseteq \mathbb{R}^n$ is called `hyper-space` *if there exists a vector $a \in \mathbb{R}^n \setminus \{0\}$ and a number $\alpha \in \mathbb{R}$ with*
$$G = \{x \in \mathbb{R}^n \mid a^T x = \alpha\}.$$

(b) A subset $H \subseteq \mathbb{R}^n$ is called `half-space` *if there exists a vector $a \in \mathbb{R}^n \setminus \{0\}$ and a number $\alpha \in \mathbb{R}$ with*
$$H = \{x \in \mathbb{R}^n \mid a^T x \leq \alpha\}.$$

(c) A subset $P \subseteq \mathbb{R}^n$ is called `polyhedron` *if there exists a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$ with*
$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\}.$$
We also write $P = P(A, b)$.

(d) A polyhedron is called `polytop` *if it is bounded, i.e., there exists a number $B \in \mathbb{R},\ B > 0$ with*
$$P \subseteq \{x \in \mathbb{R}^n \mid ||x|| \leq B\}.$$

We have various possibilities in order to describe a polyhedron.

3.6 Faces of Polyhedra

We shortly give some basic facts about the concept of faces and facets of a polyhedron which are important for the understanding of branch-and-cut algorithms. Also the definition of a vertex will be needed in the special separation algorithm we have developed for our problem. At first we define the concept of a valid inequality.

Definition 8 *Let $S \subseteq \mathbb{R}^n$, $a \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$. The inequality $a^T x \leq \alpha$ is called* `valid` *for S if*

$$S \subseteq \{x \in \mathbb{R}^n \mid a^T x \leq \alpha\}.$$

If we consider the hyper-space that is introduced by a valid inequality (which can be understood as a half-space of course) and take the intersection with a polyhedron we get a face of the polyhedron.

Definition 9 *Let $P \subseteq \mathbb{R}^n$ be a polyhedron. A set $F \subseteq P$ is called* `face` *of P if there exists a valid inequality $d^T x \leq \delta$ with*

$$F = P \cap \{ x \in \mathbb{R}^n \mid d^T x = \delta \}.$$

The face is called `proper` *if $F \neq P$ holds. F is called* `nontrivial` *if $\emptyset \neq F \neq P$ holds. If $d^T x \leq \delta$ is valid for P we call $P \cap \{x \in \mathbb{R}^n \mid d^T x = \delta\}$ the face induced by $d^T x \leq \delta$.*

For the optimal use of branch-and-cut algorithms it will be very important to know something about high dimensional faces. Therefore we come to the next

Definition 10 *Let $P = P(A, b)$ be a polyhedron. We define:*
A nontrivial face F of $P(A, b)$ is called `facet` *of $P(A, b)$ if F is not a subset of any proper face of $P(A, b)$.*

We will see in the next sections that in real problems it is often quite difficult to find exact formulas for the description of faces and facets. But we will show that it is possible to find valid inequalities from the knowledge of the vertices of a polyhedron. With special lifting techniques it is possible to get faces and even facets from these valid inequalities. We will not describe the procedure of lifting because it is very special. More information can be found in [32], [27].

Definition 11 *Let $P \subseteq \mathbb{R}^n$ be a polyhedron and F a face of P. If there exists $x \in \mathbb{R}^n$ with $F = \{x\}$ we call F vertex.*

We see that the vertices are of course faces of dimension zero. Faces of dimension one we call edges of the polyhedron.
Now we shortly describe the principle of SOS conditions and cutting plane algorithms.

3.7 Modelling piecewise linear Functions and SOS Conditions

In this section we want to give a short outline how it is possible to approximate a nonlinear function by piecewise linear functions in a mixed integer model. Let $f : \mathbb{R} \to \mathbb{R}$ be a nonlinear real function.
In order to approximate this function by a piecewise linear function on the interval $[a, b], a, b \in \mathbb{R}, a < b$ we define a set Λ of variables for the grid points and a set Y of variables that define sectors of the x-axis (see Figure 3.1). We define the set Λ as follows:

$$\Lambda = \{\lambda^1, \lambda^2, \ldots, \lambda^n\}.$$

The set Y is declared by

$$Y = \{y^1, y^2, \ldots, y^{n-1}\}.$$

Set

$$y^j = \begin{cases} 1 & \text{, if } f \text{ is approximated by conv. combin. of } f^j \text{ and } f^{j+1} \\ 0 & \text{else.} \end{cases}$$

Here the values $f^i, i \in \{1, 2, \ldots, n\}$ are the exact function values at the grid points.
Further we define for $j \in \{1, 2, \ldots, n-1\}$:

$$N(j) = \{\lambda^j, \lambda^{j+1}\}.$$

The traditional way of linearising such functions is as follows (see also Figure 3.1):

$$\begin{aligned}
\lambda^1 &\leq y^1 \\
\lambda^i &\leq y^{i-1} + y^i \qquad \forall i = 2, \ldots, n-1 \\
\lambda^n &\leq y^{n-1} \\
\textstyle\sum_{i \in \Lambda} \lambda^i &= 1 \\
\textstyle\sum_{j \in Y} y^j &= 1 \\
f &= \textstyle\sum_{i \in \Lambda} f^i \, \lambda^i \\
y^j &\in \{0, 1\} \qquad \forall j \in Y \\
\lambda^i &\geq 0 \qquad \forall i \in \Lambda.
\end{aligned}$$

This approximation idea is the traditional way known from pertinent text books, see e.g. [20], [32] or [47]. The latter one extends this formulation on the two dimensional case.
One of our aims was to develop a model which can be generalized to functions $f : \mathbb{R}^n \to \mathbb{R}$. We will see in the next chapter that the generalisation of this formulation is more complex than the generalisation of an equivalent formulation we now present.

Alternatively we can linearise our function piecewise via the following conditions:

$$\begin{aligned}
\textstyle\sum_{i \in \Lambda} \lambda^i &= 1 \\
\textstyle\sum_{j \in Y} y^j &= 1 \\
y^j &\leq \textstyle\sum_{k \in N(j)} \lambda^k \quad \forall j \in Y \\
f &= \textstyle\sum_{i \in \Lambda} f^i \, \lambda^i \\
y^j &\in \{0, 1\} \qquad \forall j \in Y \\
\lambda^i &\geq 0 \qquad \forall i \in \Lambda.
\end{aligned}$$

We see that the function f is approximated by a convex combination of exact function values at the grid points. Via the binary variables we have ensured that only $\lambda-$variables from exactly one segment do not vanish, i.e., we do not approximate f by function values which belong to grid points of different segments. In Chapter 5 we give an example why we are working with this formulation in order to model our MIP.

Let us describe another way to model this situation. We want to get rid of the binary variables (compare e.g. [17],[18],[1], [4]).
So we declare

$$\sum_{i \in \Lambda} \lambda^i = 1$$

as SOS Type 2 equation, that is

- at most two $\lambda-$variables are positive,

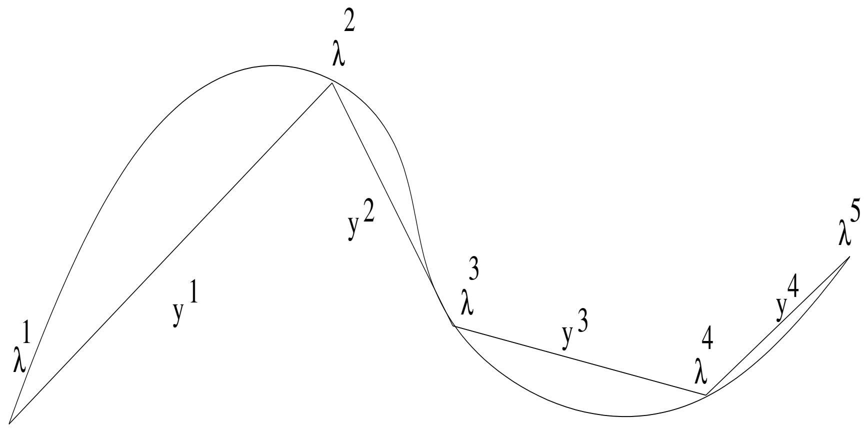

Figure 3.1: Approximation of a linear function

- if two $\lambda-$variables are positive these variables must appear consecutively.

Although this is at first glance only a reformulation we have the advantage that binary variables are no longer necessary. The additional requirements can be incorporated easily within branch-and-bound (see Section 3.8.2). We will generalise this modeling for 2- and 3-dimensional functions and to study the polyhedral consequences of the model according to our special situation.

3.8 Relaxations

In this section we give a short overview about the most common algorithms for the solution of mixed integer programs which we use for the solution of the Transient Technical Optimisation, like cutting plane or branch-and-bound algorithms. The formal description of the algorithms is taken from [13].

3.8.1 Cutting Planes

Remember the general Mixed Integer Program

$$\begin{aligned} \min\ & c^T x \\ & Ax \le b \\ & x \in \mathbb{Z}^p \times \mathbb{R}^{n-p}. \end{aligned} \tag{3.2}$$

Now we forget the integrality conditions of the variables $x_1, \dots, x_p$ and we get the so-called **linear programming relaxation**:

$$\begin{aligned} z_{\text{LP}} = \min \quad & c^T x \\ \text{s.t.} \quad & Ax \le b \\ & x \in \mathbb{R}^n. \end{aligned} \tag{3.3}$$

The linear programming relaxation is usually solved by the well known simplex algorithm.
Let $P_{\text{MIP}} := \text{conv}\{x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} : Ax \le b\}$, and $P_{\text{LP}} := \text{conv}\{x \in \mathbb{R}^n : Ax \le b\}$.
Let $x^* = (x_1^*, \dots, x_n^*)$ be an optimal solution of the linear programming relaxation. If $x^* \in \mathbb{Z}^p \times \mathbb{R}^{n-p}$ holds we obviously have solved the mixed integer problem. If not, than it is well known (see [49], [32]) that there exists a valid inequality $a^T x \le \beta$ for the polyhedron P_{MIP} that cuts off (goemetrically speaking) x^*. Because of this $a^T x \le \beta$ is called a **cutting plane**. In a cutting plane algorithm $a^T x \le \beta$ is added to (3.3) and so we get

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \le b \\ & a^T x \le \beta \\ & x \in \mathbb{R}^n, \end{aligned} \tag{3.4}$$

We continue this procedure until we get a solution of the linear programming relaxation that is in $\mathbb{Z}^p \times \mathbb{R}^{n-p}$.
The complete algorithm now becomes (see [13]):

Algorithm 12 *(Cutting Plane)*

1. *Let* $k := 0$ *and* LP^0 *the linear programming relaxation of the mixed integer program.*
2. *Solve* LP^k. *Let* $\tilde{x}^k$ *be an optimal solution.*
3. *If* $\tilde{x}^k$ *is in* $\mathbb{Z}^p \times \mathbb{R}^{n-p}$, *stop;* $\tilde{x}^k$ *is an optimal solution of the mixed integer program.*
4. *Otherwise, find a linear inequality, that is satisfied by all feasible mixed integer points, but not by* $\tilde{x}^k$.
5. *Add this inequality to* LP^k *to obtain* LP^{k+1}.
6. *Increase* k *by one and go to Step 2.*

From the previous descriptions it is clear how LP^{k+1} is calculated from LP^k.
In Chapter 5 we develop a special class of cutting planes which we have implemented in a cutting plane algorithm of the form of Algorithm 12.
A description of a huge class of well known cutting planes for several general or special problems are to be found in [27], [28], [13]. These cutting planes are already implemented in common MIP-solvers. In our test calculations we will see that our specially developed cutting planes usually improve the calculation time of the models more than these general cuts can do.

3.8.2 Branch-and-Bound and Branch-and-Cut

Branch-and-bound algorithms are well known in discrete mathematics. In this section we give a short outline how they are usually used in mixed integer programming.
With $X = \{x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} : Ax \leq b\}$ we can write the general mixed integer problem as

$$\begin{array}{rll} z_{\text{MIP}} = & \min & c^T x \\ & \text{s.t.} & x \in X, \end{array}$$

We now have the possibility (in some cases even the use of a cutting plane algorithm will not be enough for a fast solution of a mixed integer problem) to divide X into a finite number of disjoint subsets $X_1, \ldots, X_k \subset X$ with $\cup_{j=1}^k X_j = X$. Afterwards we solve the subproblems

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & x \in X_j \end{array}$$

for all $j = 1, \ldots, k$.

This procedure can be done iteratively and so we get the well known branch-and-bound tree. Summarising this basic idea we get the following algorithm (see again [13]).

Algorithm 13 *(Branch-and-Bound)*

1. *Let* L *be the list of unsolved problems. Initialize* L *with the MILP which has to be solved. Set* $U := +\infty$ *as upper bound.*
2. *Choose an unsolved problem* X_j *from the list* L *and delete it from* L.
3. *Compute the lower bound* b_{X_j} *by solving the linear relaxation. Let* $\tilde{x}_{X_j}$ *be the optimal solution, so* $b_{X_j} := c^T \tilde{x}_{X_j}$.

4. *If* $\tilde{x}_{X_j} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}$*, problem* X_j *is solved and we found a feasible solution of* X_j*; if* $U > b_{X_j}$*, set* $U := b_{X_j}$ *and delete all subproblems* X_i *with* $b_{X_i} \geq U$ *from the list.*

5. *If* $\tilde{x}_{X_j} \notin \mathbb{Z}^p \times \mathbb{R}^{n-p}$*, split problem* X_j *into subproblems and add them to the list* L*.*

6. *Go to Step 2, until the list is empty.*

Cutting plane algorithms and branch-and-bound algorithms are usually combined in order to fasten the solution time of mixed integer programs which we call a branch-and-cut algorithm.

Algorithm 14 *(Branch-and-Cut)*

1. *Let* L *be the list of unsolved problems. Initialise* L *with the mixed integer program which has to be solved. Set* $U := +\infty$ *as upper bound.*

2. *Choose an unsolved problem* X_j *from the list* L *and delete it from* L*.*

3. *Compute the lower bound* b_{X_j} *by solving the linear relaxation. Let* $\tilde{x}_{X_j}$ *be the optimal solution, so* $b_{X_j} := c^T \tilde{x}_{X_j}$*.*

4. *If* $\tilde{x}_{X_j} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}$*, problem* X_j *is solved and we found a feasible solution of* X_j*; if* $U > b_{X_j}$*, set* $U := b_{X_j}$ *and delete all subproblems* X_i *with* $b_{X_i} \geq U$ *from the list.*

5. *If* $\tilde{x}_{X_j} \notin \mathbb{Z}^p \times \mathbb{R}^{n-p}$*, look for cutting planes and add them to the linear relaxation.*

6. *Go to Step 3, until no more violated inequalities can be found or violated inequalities have too little impact in improving the lower bound.*

7. *Split problem* X_j *into subproblems and add them to the list* L*.*

8. *Go to Step 2, until the list is empty.*

There are a lot of strategies for the selection of the nodes like Best First Search, Depth First Search, Best Projection and for the selection of the variables Most Infeasibility, Pseudo-costs and Strong Branching strategies, see [13].
For the solution of the Transient Technical Optimisation we have developed a branch-and-cut algorithm which we will describe in Chapter 5.
Nearer informations about solution strategies of mixed integer programs based on cutting planes, branch-and-bound or branch-and-cut can be found in the cited text books, i.e., [7], [8], [22], [33], [34], [37], [49], [32], [47].

Let us come now to the formulation of a Mixed Integer Program for the stationary case of the Transient Technical Optimisation Problem.

Chapter 4

The Model

4.1 Introduction

In the following chapter we develop a Mixed Integer Model for the optimisation of gas networks. In Chapter 2 we have described the physical basics of the problem. Here we formulate a system of linear (in-)equalities in order to define the behaviour of valves, control valves, compressors, pipes and the other components of a gas network that we want to integrate in our model. So we have to emphasise on the combinatorial constraints (e.g. we get binary variables that indicate whether a compressor or a valve is switched on or off). The second important problem are the nonlinear components of the model. We have approached it by approximating the nonlinearities (we have already mentioned that the most important nonlinear functions in this model describe the fuel gas consumption of the compressors and the pressure loss in the pipes) by piece-wise linear functions. Since we want to solve this mixed integer program via a branch-and-cut algorithm we give a mathematical analysis of this model in Chapter 5 which belongs to the shape of the nonlinearities.

We will restrict us to the stationary case of the problem, that means that the gas flow in the segments and the pressure in the nodes is independent from time.

In this chapter we proceed as follows: First we show how to model the problem in a graph, then we introduce the most important variables and after that we formulate the conditions that are necessary in order to describe the properties of each type of segment or of each type of node. After that we will conclude with a summarising description of the whole model. Finally our preliminary computational results show the calculations of the model for a simple gas network when solving it with a general mixed integer programming solver. In Figure 4.1 we see a simple example of a gas network with a compressor, a valve and a control valve.

4.2 Basic Preliminaries of the Model

The problem of the Transient Technical Optimisation is evidently modelled in a directed graph $G = (V, E)$. The set E of segments here is partitioned in the set of compressors, the set of valves, the set of control valves, the set of pipes and the set of connections. Connections can be understood as a special kind of pipes which are very short so that they do not have any pressure loss. The set V of nodes consists of the set of intersection points of the segments, the set of sources (the gas delivering points) and the set of sinks (which are the gas demanding points of the gas network). We assume the graph to be directed since the gas flow in each segment in our model is assumed to be directed (this means we do not allow back-flow in the pipes).

We now point out the most important kinds of variables which are necessary to understand the succeeding descriptions of the several constraints of the model. At first we introduce flow variables $q_e, e \in E$ and $q_v, v \in V$. These variables describe the gas volume flow in each segment, i.e., the gas volume flow in the valves, control valves, pipes, compressors and connections or the gas flow in the nodes. Second

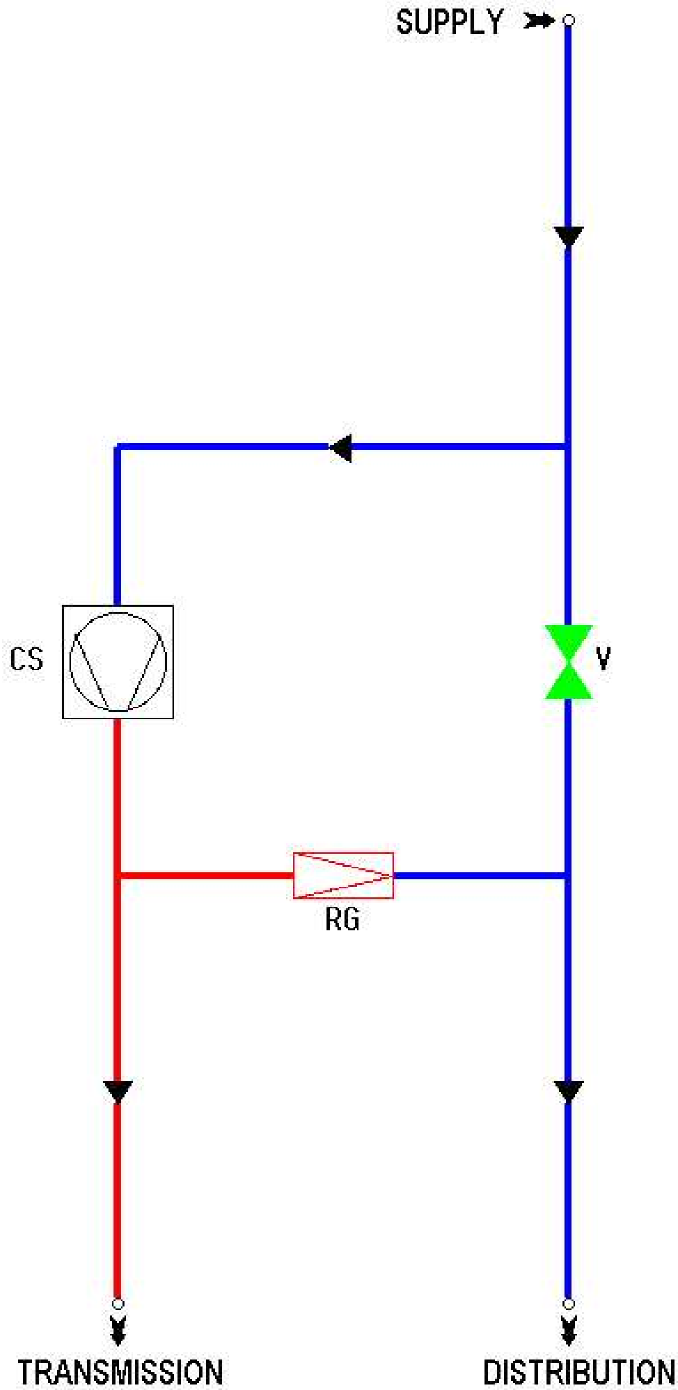

Figure 4.1: Schematic of a pipe network section with switching components: compressor **CS**, valve **V** and control valve **RG** connected by pipes, see [3]

we consider pressure variables $p_v, v \in V$. The pressure variables describe the pressure of the gas in each node, i.e., the pressure in the sources, the sinks and the pressure in the intersection points. With p_{in} and p_{out} we mark the pressure in the node of the beginning or at the end point of a certain segment. These variables are nonnegative continuous variables. Additionally to the flow and the pressure variables there are variables f and N for every compressor which describe the fuel gas consumption of the compressor and its power. These variables are very important since the compressors are driven with a fraction of the gas directed through the gas network. Our aim is to minimise this fuel gas consumption. Nowadays the fuel gas consumption of the compressors is on average about 2% of the gas flow through the compressors.

The variable N is also important since the power is connected with the fuel gas consumption and the power of the compressor has to be within certain bounds. After these continuous variables we introduce some binary variables $s \in \{0, 1\}$ which are switching variables of a valve, control valve or a compressor. It is clear that we need such kinds of variables because the valves, control valves and compressors can be switched on or off.

We remark that generally all variables and constants in this model are nonnegative and if there is a variable which values can be negative we will regard on this fact!

In the next section of this chapter we will describe all important constraints that are necessary to formulate the stationary case of the Transient Technical Optimisation Problem.

4.3 Description of the individual Constraints of the Model

For the description of the individual constraints of the model we proceed as follows:

We built up the model step by step. For each node and for each type of segment the necessary constraints are formulated separately. The notation in this section will be described as easily as it is possible, i.e., we will describe the constraint for a single segment or a single node resp. to the type of the segment or node. After we have described all principle types of constraints we will finally summarise the complete model. Because of this we will describe the needed formalism in the whole complexity for the entire model in Section 4.

So let us come now to a first description of the needed constraints. In order to emphasise the constraints they are numbered consecutively.

4.3.1 Modelling the Constraints in the Nodes

For all nodes, i.e., for all intersection nodes, all sources and all sinks the first law of Kirchhoff must be observed:

The first law of Kirchhoff deals with the balance of gas-flows in each node $i \in V$. The sum of the ingoing gas flows must be equal to the sum of the outgoing gas flows.

We can formalise the first law of Kirchhoff by introducing the gas flow variables q_e for a segment:

Let $i \in V$ be the considered node. Remember that $\delta^+(i)$ means the set of ingoing segments of node i and $\delta^-(i)$ means the set of outgoing segments of node i. Taking this formalism into account we get the following formulation of the first law of Kirchhoff:

$$\sum_{e \in \delta^+(i)} q_e \quad = \quad \sum_{e \in \delta^-(i)} q_e. \tag{4.1}$$

It is important to notice that in the node which corresponds to the endpoint of a compressor the fuel gas consumption is also observed in the Kirchhoff law (that means that the fuel gas consumption is added to the gas flow in the node at the endpoint of the compressor).

Let us describe this formula with a simple example:

In Figure 4.2 there are three ingoing segments (which may be pipes, valves, control valves, connections and compressors) and two outgoing segments. Considering the first law of Kirchhoff we get the

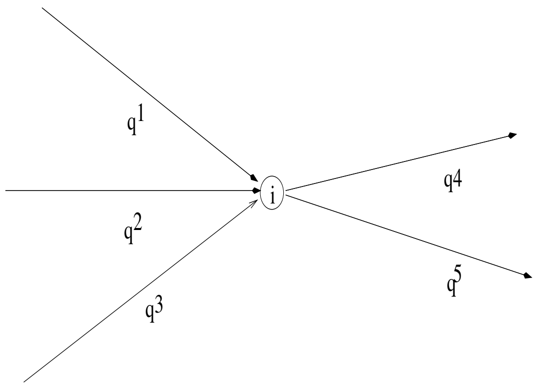

Figure 4.2: Schematic of first law of Kirchhoff

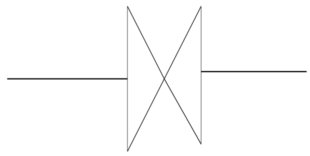

Figure 4.3: Symbol of a valve

following condition:

$$q_1 + q_2 + q_3 = q_4 + q_5.$$

After this for all nodes $i \in V$ we get lower and upper bounds for the pressure (p^{min}, p^{max}) and lower and upper bounds for the gas flow (q^{min}, q^{max}) in the nodes. So if q is the gas flow in the node and p the pressure in the node we get the following lower and upper bounds:

$$q^{min} \leq q \leq q^{max}, \tag{4.2}$$
$$p^{min} \leq p \leq p^{max}. \tag{4.3}$$

4.3.2 Modelling of a Valve

In order to model the properties of a valve we have to recognise that a valve can be closed (switched off) or open (switched on) so that we have to introduce a binary variable $s_v \in \{0, 1\}$ which describes if the valve v is open or closed. The symbol of valve in a technical description of a gas network is shown in Figure 4.3.

Let us now come to the constraints:
If the valve is closed, i.e., if $s_v = 0$, there must not be any gas flow through the valve and if the valve is open, i.e., if $s_v = 1$, the gas flow is bounded from above by a maximal flow rate q^{max} and so we get:

$$q \leq q^{max} s_v. \tag{4.4}$$

We get a second analogously constraint. Is $s_v = 1$ the gas flow also shall be bounded from below by a minimum flow rate q^{min} and so we get:

$$q \geq q^{min} s_v. \tag{4.5}$$

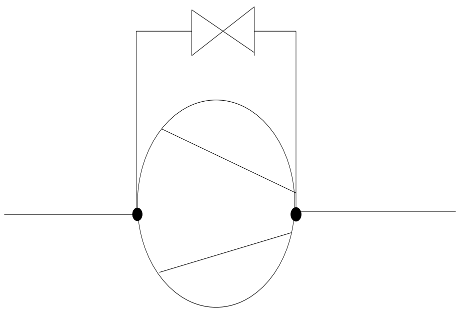

Figure 4.4: Valve as bypass valve of a compressor

These two constraints ensure that if $s_v = 0$ there is now gas flow through the valve and if $s_v = 1$ the gas flow fulfils the condition:

$$q^{min} \leq q \leq q^{max}.$$

We usually set $q^{min} = 0$. There are two additional constraints for a valve that only must hold in a special case. Some valves are only needed in a direct connection with a compressor. Since a compressor also can be switched on or off, what is the consequence if the compressor is switched off? In this case the gas cannot flow through the compressor. So a valve is built parallel to the compressor which we call a bypass valve. If the compressor is switched off the gas flows through the valve.
Figure 4.4 describes the situation.
A bypass valve of a compressor has to fulfil the following additional inequalities:

$$p_{in} - p_{out} \geq dp^{max}(s_v - 1). \tag{4.6}$$

This inequality, where p_{in} is the pressure at the node of the beginning of the valve and the compressor, p_{out} is the pressure at the node of the end of the valve and the compressor and dp^{max} means a nonnegative real number which describes the minimal pressure difference between these two nodes (it is clear that dp^{max} is the difference between the maximal pressure value in the node at the end of the valve and the minimal pressure value in the node at the beginning of the valve), describes the following:
If the valve is on, i.e., $s_v = 1$, which also means that the switching variable of the compressor is zero (see the subsection of the compressor) than $p_{in} - p_{out} \geq 0$ holds. In the case $s_v = 0$ we see that the constraint has no consequence.
The following constraint

$$p_{in} - p_{out} \leq 0 \tag{4.7}$$

is regarded closely to the last constraint. Since the valve is a bypass valve to a compressor and a compressor is built in order to increase the gas pressure again is obviously that $p_{in} - p_{out} \leq 0$ holds. In the case $s_v = 1$ we conclude by the previous constraint the equality $p_{in} = p_{out}$.

As it is easy to see there are also some other constraints in this case regarding the switching variables of the compressor and its bypass valve or the gas flow through a compressor. We will mention this constraint when we are describing the constraints of a compressor.

Generally we introduce the following two inequalities for valves:

$$Ms_v - p_{in} + p_{out} \leq M. \tag{4.8}$$

Here we can take $M = p_{out}^{max} - p_{in}^{min}$ where p_{out}^{max} is the maximal pressure in the node at the end of the valve and p_{in}^{min} is the minimal pressure in the node at the beginning of the valve.

$$Ms_v + p_{in} - p_{out} \leq M. \tag{4.9}$$

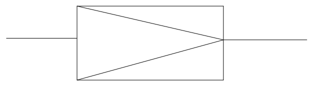

Figure 4.5: Symbol of a control valve

In this inequality we can take $M = p_{in}^{max} - p_{out}^{min}$ where p_{in}^{max} is the maximal pressure in the node at the beginning of the valve and p_{out}^{min} is the minimal pressure in the node at the end of the valve.
We remark that these two inequalities are part of the MIR (Mixed-Integer-Rounding) cuts (see e.g. [32], [28] or [27]) that can be calculated from constraints (4.8) and (4.9).

4.3.3 Modelling of a Control Valve

Control valves in principle have the same properties as valves. The basic difference between a valve and a control valve is that a control valve can control down the pressure at the end of it. The control valve can be switched on and switched off again. The switching variable here may be $s_r \in \{0, 1\}$.
The usual symbol for a control valve is shown in Figure 4.5. So the properties of control valves can be modelled as follows:
First we get two constraints which are obviously analogous to the situation for a valve:

$$q \leq q^{max} s_r, \tag{4.10}$$

$$q \geq q^{min} s_r. \tag{4.11}$$

We usually set $q^{min} = 0$. There are two more constraints that are needed in order to describe the behaviour of the control valve:
We want to model that if the control valve is on , i.e, $s_r = 1$, the control valve shall manage the pressure. Because of this we introduce a positive constant dp^{max} which describes the maximal value how much the control valve can regulate down the pressure (dp here means "difference of pressure" - since there is no variable d in our model there is no danger of confusion). If the control valve is switched off, i.e, $s_r = 0$, we introduce a relatively big constant M (a so called "big M") such that in this case the pressure in the node at the beginning of the control valve (p_{in}) and the pressure in the node at the end of the control valve (p_{out}) is arbitrary. We can formulate this situation with the following constraint:

$$p_{in} - p_{out} \leq dp^{max} s_r + M(1 - s_r). \tag{4.12}$$

Now we have to model an analogous situation because if the control valve is on it shall control down the pressure at least by a minimal value dp^{min}. Again if the control valve is switched off p_{in} and p_{out} shall be arbitrary and so we add the following constraint:

$$p_{in} - p_{out} \geq dp^{min} s_r - M(1 - s_r). \tag{4.13}$$

If we consider constraints (4.12) and (4.13) together we see that the introduction of the term $M(1 - s_r)$ is necessary. If we would not have done this we would get in the case $s_r = 0$ the condition $p_{in} = p_{out}$ which is of course not a correct formulation.
We remark that a control valve never is used as a bypass valve to a compressor.

4.3.4 Modelling the Properties of a Connection

We remember that a connection can be understood as a special pipe which is so short that there is no pressure loss.

So we only have to consider lower (q^{min}) and upper (q^{max}) bounds for the gas flow q in a connection and we get the constraint:

$$q^{min} \leq q \leq q^{max}. \tag{4.14}$$

Since there is no pressure loss we get the constraint

$$p_{in} = p_{out}. \tag{4.15}$$

4.3.5 Modelling the Properties of a Pipe

At first there are lower (q^{min}) and upper (q^{max}) bounds for the gas flow q in a pipe. So we get the constraint:

$$q^{min} \leq q \leq q^{max}. \tag{4.16}$$

The description of the pressure loss in the pipes is relatively complex. The reason for this is that the pressure loss cannot be described by a simple linear function.

We shortly remember the already known structure of this nonlinear function.
Remember the momentum equation (we describe it here for cylindrical pipes) in the notation of Chapter 2:

$$\frac{\partial p}{\partial x} + g\rho\frac{\partial h}{\partial x} + \frac{\lambda|v|v}{2D}\rho + \frac{1}{A}\frac{\partial M}{\partial t} + \frac{\partial(\rho v^2)}{\partial x} = 0.$$

We remember that this equation simplifies for the stationary case and for horizontal pipes to:

$$p_{out}^2 = p_{in}^2 - \mathrm{ff}\, q\, |q|, \tag{*}$$

where holds:

$$\mathrm{ff} = \mathrm{ff}(p_{out},\, p_{in}).$$

Since we cannot use (*) in a Mixed Integer Model we want to build a piecewise linear approximation of (*).

Some calculations showed us that we can assume ff to be constant and so in principle we get the situation that we want to approximate a nonlinear function $p_{out} = p_{out}(p_{in}, q)$ where p_{out} is the pressure at the end of the pipe, p_{in} is the pressure at the beginning of the pipe and q is the gas flow through the pipe. We define some grid points of the form (p_{in}, q) and associate nonnegative weightings λ for each grid point. The pressure loss in a pipe is visualised in Figure 4.6. In Figure 4.6 the grid of the values (p_{in}, q) is equidistant.
So our approximation of the pressure loss in pipes is done in the following way:

Define a *decomposition* of the two-dimensional manifold (function) $p_{out} = p_{out}(p_{in}, q)$. These triangulations are done by a triangulation of the domain of p_{out} in triangles, that means:

- *Definition* of a set Λ_p of two-dimensional grid points.
- *Linearisation* of the function p_{out} for each element of the set Y_p of triangles between the grid points.

Subindex p here stands for a pipe. The principle situation is described in Figure 4.7.

Modelling this we get the following variables:

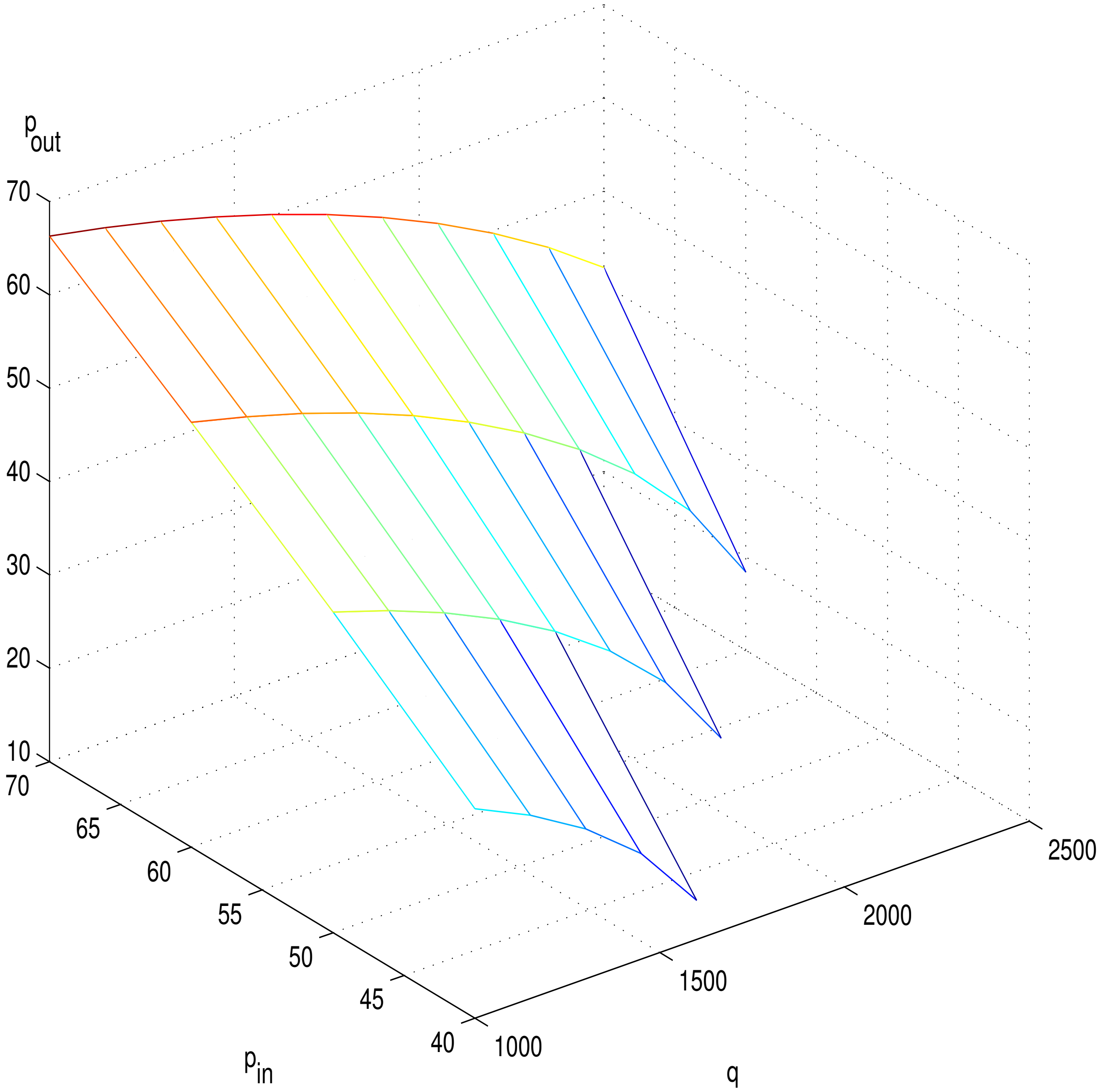

Figure 4.6: Piecewise linearisation of the pressure loss in a pipe

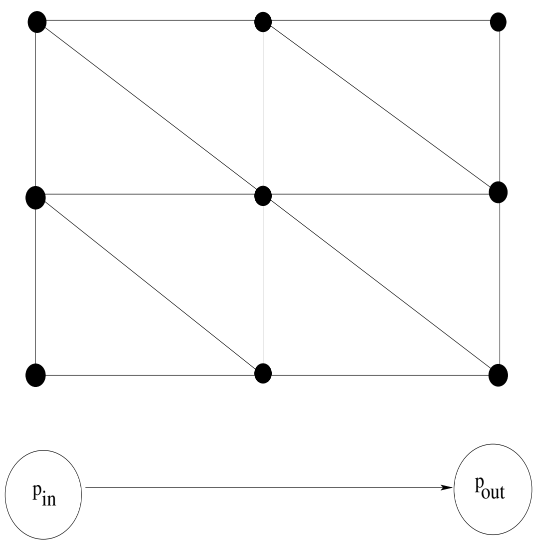

Figure 4.7: Typical triangulation of the pressure loss in a pipe

- There is a variable λ^i for each grid point $i \in \Lambda_p$ (it describes the fraction of a special grid point of the linearisation).
- There is a variable $y^j \in \{0, 1\}$ for each triangle $j \in Y_p$.

Clearly we define for $j \in Y_p$:

$$y^j = \begin{cases} 1 & \text{, if the value of } p_{out} \text{ is approximated by } \lambda-\text{variables of triangle } j, \\ 0 & \text{, else.} \end{cases}$$

As an example consider Figure 4.8, which shows the variables introduced for the example in Figure 4.7.
With this preliminary descriptions we now can describe all the inequalities that define the piecewise linear approximation of the nonlinear function of pressure loss in a pipe.
Let for $i \in \Lambda_p$ (for the considered pipe) the numbers p^i_{in} and q^i be the values of the pressure in the node at the beginning of the pipe and the gas flow for the 2−dimensional grid points and p^i_{out} be the value of the nonlinear function for the grid point, i.e., it holds for $i \in \Lambda_p$

$$p^i_{out} = p_{out}(p^i_{in}, q^i).$$

For $j \in Y_p$ let $N(j)$ be the set of $\lambda-$variables that belong to the considered triangle.
As an easy example we see that in Figure 4.8 holds $N(1) = \{1, 5, 6\}, N(2) = \{1, 2, 6\}$, $N(3) = \{2, 6, 7\}, \dots$.

So we get as a first constraint that the sum of all $\lambda-$variables of the pipe must be equal to one since we want to linearise the nonlinear function piecewise by convex combinations of grid points and so we get

$$\sum_{i \in \Lambda_p} \lambda^i = 1, \qquad \lambda^i \geq 0. \tag{4.17}$$

The next constraint describes that exactly one triangle is chosen in order to linearise the nonlinear function. Remember that the variables $y^i, i \in Y_p$, are binary variables so that the following constraint implies that exactly one triangle is chosen.

$$\sum_{j \in Y_p} y^j = 1, \qquad y^j \in \{0, 1\}. \tag{4.18}$$

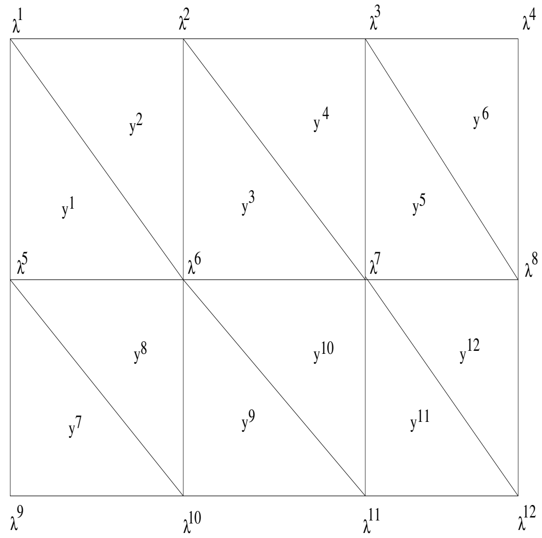

Figure 4.8: $\lambda-$ and $y-$variables in a triangulation of the pressure loss in a pipe

The constraints (4.17) and (4.18) are not enough because it is not yet ensured that the nonlinear function is only approximated by grid points which belong to exactly one triangle. But if we remember that we have defined the sets $N(j)$ for all $j \in Y_p$ we can formulate this by introducing the following constraint:

$$y^j \leq \sum_{k \in N(j)} \lambda^k \qquad \forall j \in Y_p. \tag{4.19}$$

Additionally to the constraints (4.17), (4.18), (4.19) we now calculate convex combinations from the grid points (that fulfil (4.17), (4.18), (4.19)). The convex combination according to the pressure in the node at the beginning of the pipe is modelled by the constraint:

$$p_{in} = \sum_{i \in \Lambda_p} p_{in}^i \ \lambda^i. \tag{4.20}$$

The convex combination according to the gas flow in the pipe analogously is modelled by:

$$q = \sum_{i \in \Lambda_p} q^i \ \lambda^i. \tag{4.21}$$

The last constraint which is necessary in order to complete the piecewise linearisation of the pressure loss in the pipes describes the linearisation of the value of the function, i.e., the pressure at the end of the pipe and so we get the constraint:

$$p_{out} = \sum_{i \in \Lambda_p} p_{out}^i \ \lambda^i. \tag{4.22}$$

With these conditions it is ensured that we get a piecewise linear approximation of the pressure loss function in a pipe. It is easy to see that this is a generalisation of Section 3.7 in Chapter 3.
In our test computations we noticed that the addition of the constraint

$$p_{in} - p_{out} \geq 0. \tag{4.23}$$

accelerates the calculations.

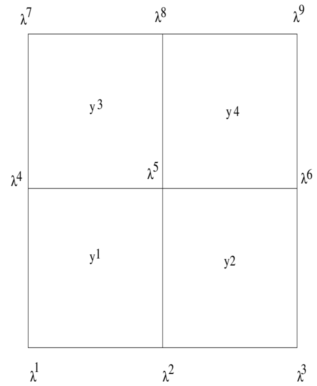

Figure 4.9: Example for inequalities connecting λ- and y-variables in case of rectangles

In Chapter 2 we stated that we take an alternative formulation for the piecewise approximation of non-linear functions to the well known formulations in [20], [32]. We give two simple examples in order to give reasons for this decision. In Figure 4.9 we show a small discretisation in rectangles (we give this example since our first discretisations have been implemented in this way). The inequalities in our formulation are:

$$\begin{aligned} y^1 &\leq \lambda^1 + \lambda^2 + \lambda^4 + \lambda^5 \\ y^2 &\leq \lambda^2 + \lambda^3 + \lambda^5 + \lambda^6 \\ y^3 &\leq \lambda^4 + \lambda^5 + \lambda^7 + \lambda^8 \\ y^4 &\leq \lambda^5 + \lambda^6 + \lambda^8 + \lambda^9 \end{aligned}$$

The inequalities connecting λ and y-variables for the formulation in [20],[32] are:

$$\begin{aligned} \lambda^1 &\leq y^1 \\ \lambda^2 &\leq y^1 + y^2 \\ \lambda^3 &\leq y^2 \\ \lambda^4 &\leq y^1 + y^3 \\ \lambda^5 &\leq y^1 + y^2 + y^3 + y^4 \\ \lambda^6 &\leq y^2 + y^4 \\ \lambda^7 &\leq y^3 \\ \lambda^8 &\leq y^3 + y^4 \\ \lambda^9 &\leq y^4 \end{aligned}$$

Figure 4.10 shows the same discretisation as Figure 4.9 with triangles.

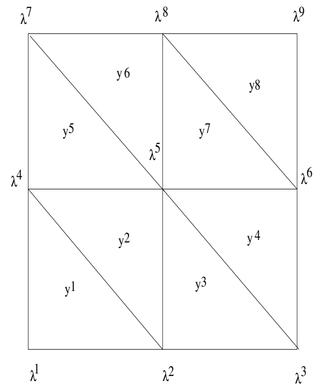

Figure 4.10: Example for inequalities connecting λ- and y-variables in case of triangles

The inequalities in our formulation are:

$$\begin{array}{rcl} y^1 & \leq & \lambda^1 + \lambda^2 + \lambda^4 \\ y^2 & \leq & \lambda^2 + \lambda^4 + \lambda^5 \\ y^3 & \leq & \lambda^2 + \lambda^3 + \lambda^5 \\ y^4 & \leq & \lambda^3 + \lambda^5 + \lambda^6 \\ y^5 & \leq & \lambda^4 + \lambda^5 + \lambda^7 \\ y^6 & \leq & \lambda^5 + \lambda^7 + \lambda^8 \\ y^7 & \leq & \lambda^5 + \lambda^6 + \lambda^8 \\ y^8 & \leq & \lambda^6 + \lambda^8 + \lambda^9 \end{array}$$

The inequalities connecting λ and y-variables for the formulation in [20],[32] are:

$$\begin{array}{rcl} \lambda^1 & \leq & y^1 \\ \lambda^2 & \leq & y^1 + y^2 + y^3 \\ \lambda^3 & \leq & y^3 + y^4 \\ \lambda^4 & \leq & y^2 + y^5 \\ \lambda^5 & \leq & y^2 + y^3 + y^4 + y^5 + y^6 + y^7 \\ \lambda^6 & \leq & y^4 + y^7 + y^8 \\ \lambda^7 & \leq & y^5 + y^6 \\ \lambda^8 & \leq & y^6 + y^7 + y^8 \\ \lambda^9 & \leq & y^8 \end{array}$$

These two examples easily can be generalized.
We note that the advantage of our formulation is that we little depend on the geometric structure of the considered discretisation. So the implementation of the inequalities becomes easier for our formulation. But we add that one disadvantage of our formulation is that in complexer discretisations we need more inequalities than the standard formulation. Using SOS Type 2 formulation and the facts we develop in Chapter 5 the two formulations become equivalent.

As a further side remark we add the following note: If we select a certain triangle with the values $(p_{in}^k, q^k)^T$ for the first corner, $(p_{in}^l, q^l)^T$ for the second corner and $(p_{in}^m, q^m)^T$ for the third corner we get for $\lambda^k, \lambda^l, \lambda^m$ that define the convex-combination of a point $(p_{in}, q)^T$ which lies in the triangle:

$$\begin{aligned}
\lambda^k &= \frac{p^l_{in}q - p^m_{in}q - p_{in}q^l + p^m_{in}q^l + p_{in}q^m - p^l_{in}q^m}{p^l_{in}q^k - p^m_{in}q^k - p^k_{in}q^l + p^m_{in}q^l + p^k_{in}q^m - p^l_{in}q^m} \\
\lambda^l &= \frac{-p^k_{in}q + p^m_{in}q + p_{in}q^k - p^m_{in}q^k - p_{in}q^m + p^k_{in}q^m}{p^l_{in}q^k - p^m_{in}q^k - p^k_{in}q^l + p^m_{in}q^l + p^k_{in}q^m - p^l_{in}q^m} \\
\lambda^m &= \frac{-p^k_{in}q + p^l_{in}q + p_{in}q^k - p^l_{in}q^k - p_{in}q^l + p^k_{in}q^l}{-p^l_{in}q^k + p^m_{in}q^k + p^k_{in}q^l - p^m_{in}q^l - p^k_{in}q^m + p^l_{in}q^m}
\end{aligned}$$

With

$$\alpha = p^l_{in}q^k - p^m_{in}q^k - p^k_{in}q^l + p^m_{in}q^l + p^k_{in}q^m - p^l_{in}q^m$$

and

$$\begin{aligned}
\beta^k &= p^l_{in} - p^m_{in} \\
\gamma^k &= q^m - q^l \\
\delta^k &= p^m_{in}q^l - p^l_{in}q^m \\
\beta^l &= p^m_{in} - p^k_{in} \\
\gamma^l &= q^k - q^m \\
\delta^l &= p^k_{in}q^m - p^m_{in}q^k \\
\beta^m &= p^k_{in} - p^l_{in} \\
\gamma^m &= q^l - q^k \\
\delta^m &= p^l_{in}q^k - p^k_{in}q^l
\end{aligned}$$

we can write $\lambda^k, \lambda^l, \lambda^m$ as

$$\begin{aligned}
\lambda^k &= \frac{1}{\alpha}(\beta^k q + \gamma^k p_{in} + \delta^k) \\
\lambda^k &= \frac{1}{\alpha}(\beta^l q + \gamma^l p_{in} + \delta^l) \\
\lambda^k &= \frac{1}{\alpha}(\beta^m q + \gamma^m p_{in} + \delta^m)
\end{aligned}$$

That means we approximate for *constant* ff (as a first approximation) the value $p_{out} = \sqrt{(p_{in})^2 - \text{ff}q^2}$ by the value

$$p^{approx}_{out} = \lambda^k \sqrt{(p^k_{in})^2 - \text{ff}(q^k)^2} + \lambda^l \sqrt{(p^l_{in})^2 - \text{ff}(q^l)^2} + \lambda^m \sqrt{(p^m_{in})^2 - \text{ff}(q^m)^2}.$$

Since we know that the pressure loss function is concave (because the square root function is concave and monotone and the function $f : \mathbb{R}^2 \to \mathbb{R} : f(x, y) = x^2 - ay^2$ with $a > 0$ is concave) we get

$$p^{approx}_{out} \leq p_{out}.$$

That means we approximate the pressure loss with a little underevaluation. But in every iteration we can calculate the absolute deviation of the pressure value and so we are always informed about the differences between approximated and real value at every iteration. In our test calculations the differences always have been very small.

The consequence for our model is that the fuel gas consumption of the compressors will be somewhat higher than in reality which is all right for our optimisation problem (that means we will never calculate a solution which is better than the real optimum - a somewhat pessimistic optimisation).

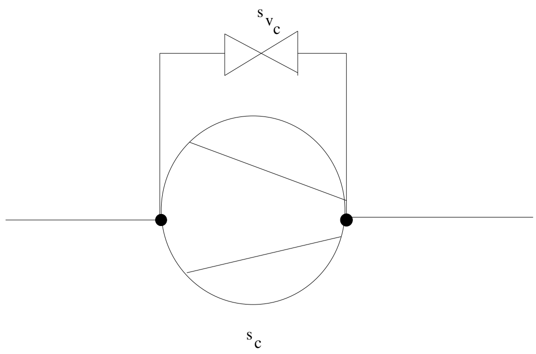

Figure 4.11: A compressor with bypass valve

4.3.6 Modelling the Properties of a Compressor

The constraints in order to formulate the properties of a compressor are quite multifaceted but we will see that all the constraints are principally known from the constraints we have formulated until now.

We can divide the constraints for modelling a compressor into two parts:
First we formulate some basic inequalities and after that we describe how to get a piecewise linearisation of the nonlinear function which describes the gas flow consumption of a compressor. This piecewise linearisation is quite analogous to the piecewise linearisation of the pressure loss in a pipe that we have discussed in the previous subsection.

The typical situation is described in Figure 4.11:
A compressor (with switching variable $s_c \in \{0,1\}$) which has to increase the pressure of the gas if necessary is constructed parallel to its bypass valve (with switching variable $s_{v_c} \in \{0,1\}$). So the gas flows through the compressor if the gas pressure must be increased and the gas flows through the bypass valve if the compressor is switched off.

Let us now come to the basic (in-)equalities for modelling a compressor:
Analogously to the case of a valve or a control valve we get two constraints that describe the gas flow through the compressor:

$$q \leq q^{max} s_c, \tag{4.24}$$

$$q \geq q^{min} s_c. \tag{4.25}$$

The next constraint is combinatorial nature and we have mentioned it implicitly several times. Either the compressor is open (binary switching variable s_c) or the bypass valve (binary switching variable s_{v_c}) is open and so we get:

$$s_c + s_{v_c} = 1. \tag{4.26}$$

Since a compressor has to compensate the pressure loss in the pipes the gas pressure at the beginning of the compressor (p_{in}) must be lower or equal to the gas pressure at the end of the compressor (p_{out}) which is modelled by the constraint

$$p_{in} - p_{out} \leq 0. \tag{4.27}$$

The fuel gas consumption f which is zero if the compressor is switched off should not be bigger as a maximal fuel gas consumption f^{max} (a constant of our model for each compressor) if the compressor is switched on and so we get the following constraint:

$$f \leq f^{max} s_c. \tag{4.28}$$

Additionally we get two more technical constraints regarding the power of a compressor:
If the compressor is closed, i.e., if $s_c = 0$, the power N of the compressor must be zero and if the compressor is open, i.e., if $s_c = 1$, the power shall be bounded from above by a maximal power rate N^{max} and so we get:

$$N \leq N^{max} s_c. \tag{4.29}$$

Analogously we get a second constraint. If $s_c = 1$ the power also shall be bounded from below by a minimum power rate N^{min} of the compressor and so we get:

$$N \geq N^{min} s_c. \tag{4.30}$$

In this model only the ideal compressor is embraced.
Now we consider the second important nonlinear function in our model that we have already described in Chapter 2:
The fuel gas consumption of the ideal compressor is a nonlinear function of the form $f = f(p_{in}, p_{out}, q)$ where p_{in} is the gas pressure in the node at the beginning of the compressor, p_{out} is the gas pressure in the node at the end of the compressor and q the gas flow through the compressor.

We will shortly remember the form of this function: We start from the adiabatic height H_{ad} of a compressor

$$H_{ad} = \frac{\kappa}{\kappa - 1} \frac{z_{in} R T_{in}}{g\, m} \{ (\frac{p_{out}}{p_{in}})^{\frac{\kappa-1}{\kappa}} - 1 \}.$$

Remember that κ (isontropic exponent), g (gravity constant), R (gas constant) and m (molecular mass of the gas) are physical or gas constants. z_{in} and T_{in} are calculated values for the gas in the node at the beginning of the compressor. The formula of the fuel consumption f is then given by

$$f = \frac{N\, b}{1000\, H_u}$$

with:

$$\begin{aligned} N_{th} &= \frac{g}{3600}\, H_{ad}\, \rho_0\, q, \\ N &= \frac{N_{th}}{\eta_{ad}}. \end{aligned}$$

Remember that N_{th} means the theoretical power of the compressor, η_{ad} the adiabatic efficiency of the compressor, ρ_0 the norm density of the gas, b the specific fuel gas consumption of the compressor and H_u the lower heat rate of the gas. These values are constants.

Let us formulate now the constraints for the compressor:
From our recapitulation it is clear that there is a direct connection between the power N and the gas flow consumption f of the compressor. So we introduce a constraint

$$f = \gamma N \tag{4.31}$$

to our model. γ is a constant which can be calculated from the complete formula for the gas consumption f of the compressor and it holds

$$\gamma = \frac{b}{1000\, H_u}.$$

It is important to explain the sense of constraint (4.28) and (4.29). Because of constraint (4.31) one could think that one of the constraints (4.28) or (4.29) could be omitted. Unfortunately it is better to have both constraints. The power of the compressor is bounded from above by N_{max} but it could be that in this case the fuel gas consumption could increase too much so that it is better to have the possibility to bound the gas flow consumption by f_{max}.

Because of constraint (4.31) we now can decide if we want to linearise the power N or the flow consumption f. Since the gas flow consumption of the sum of all compressors is to be minimized at the end our approach is to find a piecewise linear approximation of fuel consumption f. The linear approximation of the fuel consumption f is done analogously like the linear approximation of the pressure loss in a pipe.

Since we get the situation that we want to approximate a nonlinear function $f = f(p_{in}, p_{out}, q)$ (where p_{out} is the pressure in the node at the end of the compressor, p_{in} the pressure in the node at the beginning of the compressor, q is the gas flow through it and f is the fuel gas consumption) we define some grid points of the form (p_{in}, p_{out}, q) and associate again nonnegative $\lambda-$variables as "weightings" for each grid point.

So the approximation of the fuel gas consumption of compressors is done (compare the approximation of the pressure loss function) by the following steps: Define a *triangulation (decomposition)* of the three-dimensional manifold (function) $f = f(p_{in}, p_{out}, q)$. These triangulations e.g. can be done by a triangulation of the domain of f in cubes or in tetrahedrons, that means:

- *Definition* of a set Λ_c of three-dimensional grid points.
- *Linearisation* of the function f for each element of the set Y_c of cubes/tetrahedra between the grid points.

Subindex c here stands for a compressor.

The advantage of tetrahedra is that an approximation of an inner point of a tetrahedron is a *unique* convex combination of its corners.
Considering our last ventilations we get the following variables:

- There is a variable λ^i for each grid point $i \in \Lambda_c$.
- There is a variable $y^j \in \{0, 1\}$ for each cube/tetrahedron $j \in Y_c$.

We define (compare again the description of the linearisation of the pressure loss in a pipe) for $j \in Y_c$:

$$y^j = \begin{cases} 1 & \text{, if the value of } f \text{ is approximated by } \lambda-\text{variables of cube/tetrahedron } j \\ 0 & \text{, else.} \end{cases}$$

With these preliminary descriptions we now can describe all inequalities that are needed to linearise the nonlinear function of the fuel gas consumption of a ideal compressor. The description is quite the same as we have done it for the pressure loss in a pipe. We only additionally have to consider that a compressor can be switched on or switched off such that the constraints have to be a little modified.

Let for $i \in \Lambda_c$ for the considered compressor p^i_{in}, p^i_{out} and q^i be the values of the $3-$dimensional grid points and f^i the value of the nonlinear function in the grid point, i.e., it holds for $i \in \Lambda_p$:

$$f^i = f(p^i_{in}, p^i_{out}, q^i).$$

For $j \in Y_c$ let $N(j)$ the set of $\lambda-$variables that belong to the considered cube or tetrahedron.
It is clear that if we built our decomposition of cubes it holds $|N(j)| = 8$ for all $j \in Y_c$ and if we triangulate the domain of f into tetrahedrons we get $|N(j)| = 4$ for all $j \in Y_c$.
So we get as a first constraint that the sum of all $\lambda-$variables of the compressor must be equal to one if the compressor is switched on and zero if the compressor is switched off (since we want to linearise the

nonlinear function piecewise by convex combinations of grid points only if the compressor is switched on) and so we get

$$\sum_{i \in \Lambda_c} \lambda^i = s_c, \qquad \lambda^i \geq 0. \tag{4.32}$$

The next constraint describes that exactly one cube/tetrahedron is chosen in order to linearise the nonlinear function if the compressor is switched on. Remember that the variables $y^i, i \in Y_c$ are binary variables so that the constraint implies that exactly one cube/tetrahedron in this case is chosen.

$$\sum_{j \in Y_c} y^j = s_c, \qquad y^j \in \{0, 1\}. \tag{4.33}$$

Like in the situation of the pressure loss function for pipes the constraints (4.32) and (4.33) are not enough because it is not yet ensured that the nonlinear function is only approximated by grid points that belong to exactly one cube/tetrahedron. But if we remember that we have defined the sets $N(j)$ for all $j \in Y_c$ we can formulate this by introducing the following constraint (we remark that this inequality is correct formulated as well as the compressor is switched on or off):

$$y^j \leq \sum_{k \in N(j)} \lambda^k \qquad \forall j \in Y_c. \tag{4.34}$$

Additionally to the constraints (4.32), (4.33), (4.34) we now calculate convex combinations from the grid points (that fulfil (4.32), (4.33), (4.34)). The convex combination according to the pressure in the node at the beginning of the compressor is modelled by the following constraint where p_{in}^h is a nonnegative auxiliary variable:

$$p_{in} = \sum_{i \in \Lambda_c} p_{in}^i \ \lambda^i + p_{in}^h. \tag{4.35}$$

We remark that if $s_c = 1$, i.e., the compressor is switched on (4.35) must reduce to $p_{in} = \sum_{i \in \Lambda_c} p_{in}^i \lambda^i$. Is $s_c = 0$, i.e., the compressor is switched off (4.35) must reduce to $p_{in} = p_{in}^h$. Because of this we introduce the following constraint which implies $p_{in}^h = 0$ if the compressor is switched on:

$$p_{in}^{max} \, (s_c - 1) + p_{in}^h \leq 0. \tag{4.36}$$

p_{in}^{max} is a constant that ensures that (4.36) is of no relevance if $s_c = 0$. So all conditions are fulfilled. We mention that the auxiliary variable is important because without this variable we would get $p_{in} = 0$ if the compressor is switched off in (4.35) and this would be wrong of course. p_{in} in this case can be calculated from the ingoing segments of the compressor.
The convex combination according to the gas pressure at the end of the compressor now clearly is modelled by an analogous constraint. Here we need a nonnegative auxiliary variable p_{out}^h in order to get a correct formulation:

$$p_{out} = \sum_{i \in \Lambda_c} p_{out}^i \ \lambda^i + p_{out}^h. \tag{4.37}$$

In order to be correct we additionally introduce the following constraint with a constant p_{out}^{max}:

$$p_{out}^{max} \, (s_c - 1) + p_{out}^h \leq 0. \tag{4.38}$$

Now p_{out}^h is zero if the compressor is switched on and $p_{out} = p_{out}^h$ if it is switched off and so the formulation is correct.
The convex combination according to the gas flow in the compressor becomes:

$$q = \sum_{i \in \Lambda_c} q^i \ \lambda^i. \tag{4.39}$$

Here no auxiliary variable is necessary since the gas flow through the compressor is zero if the compressor is switched off because of (4.24) and (4.25).

The last constraint which is necessary in order to complete the piecewise linearisation of the gas flow consumption in the compressor now describes the linearisation of the value of the function, i.e., the gas flow consumption f of the compressor and so we get the constraint:

$$f = \sum_{i \in \Lambda_c} f^i \ \lambda^i. \tag{4.40}$$

Here also no auxiliary variable is necessary since the fuel gas consumption through the compressor is zero if the compressor is switched off because of (4.28).

4.4 Summary of the whole Model

In the previous section we have described for each type of segment and for each type of node by means of a single element of each type the constraints that are necessary in order to built up a stationary Mixed Integer Model for the Optimisation of Gas Networks. In this section we want to summarise the whole mixed integer linear model (MILP). Because of this we have to introduce some definitions and a little bit more formalism because we now cannot longer deal with a single segment or node of a special type. Nevertheless the reader will easily recognise that all the constraints are the same.

4.4.1 Definitions

The following basic sets define the essential types of segments and nodes:

V	Set of nodes
E	Set of segments
E_P	Set of pipes
E_C	Set of compressors
E_V	Set of valves
E_R	Set of control valves
E_A	Set of connections

It holds:

$$E = E_P \ \dot{\cup}\ E_C \ \dot{\cup}\ E_V \ \dot{\cup}\ E_R \ \dot{\cup}\ E_A.$$

$\dot{\cup}$ stands for the disjoint union of two sets. v denotes an element of V and e an element of E. An index e denotes a variable or constant belonging to the element e. For example s_e is the switching variable of the compressor e if $e \in E_C$. We remark that in this case s_{v_e} $(v_e \in E_V)$ denotes the switching variable of the bypass valve of compressor $e, e \in E_C$ (we write s_{v_e} since we define the valve as a bypass valve of compressor e). Sometimes we will write $e = (u, v)$ where u deals with the node at the beginning of segment e and v means the node at the end of segment e. This is especially the case when we are dealing with the piecewise linearisation of a nonlinear function (e.g. $p^i_{e,u}$ means the pressure at the beginning of a special segment e at a grid point i).
For $e \in E_P$ we denote with N^j_e the set of grid points that are belonging to triangle j of the triangulation. Analogously for $e \in E_C$ we denote with N^j_e the set of grid points that are belonging to cube/tetrahedron j. In both cases p_u means the pressure in the node at the beginning of the pipe, p_v the pressure in the node at the end of the pipes and so on.

In order to give the formal description of the whole model and for a summary of the last declarations we give the following table:

Variables of the model

q_e	flow variable of a segment $e \in E$
s_e	switching variable of of a segment $e \in E \setminus (E_P \cup E_A)$
p^e_{in}	pressure variable at the node at the beginning of a control valve
p^e_{out}	pressure variable at the node at the end of a control valve
s_{v_e}	switching variable of bypass valve of compressor $e \in E_C$
f_e	fuel consumption of compressor $e \in E_C$
N_e	power of compressor $e \in E_C$
q_v	flow in node $v \in V$
p_v	pressure in node $v \in V$
λ^i_e	$\lambda-$variable for linearisation of pipe $e \in E_P$ or compressor $e \in E_C$
y^j_e	$y-$variable for linearisation of pipe $e \in E_P$ or compressor $e \in E_C$
p_u	pressure variable at the node at the beginning of a pipe or compressor
p_v	pressure variable at the node at the end of a pipe or compressor
p^h_u	auxiliary variable for the pressure at the node at the beginning of a compressor
p^h_v	auxiliary variable for the pressure at the node at the end of a compressor

Constants and other values

q^{max}_e	maximal flow of segment $e \in E$
q^{min}_e	minimal flow of segment $e \in E$
dp^{max}_e	maximal pressure regulation of control valve $e \in E_R$
$dp^{min}_{v_e}$	minimal pressure regulation of control valve $e \in E_R$
M_e	constant “big M” depending on segment $e \in E_R \cup E_V$
f^{max}_e	maximal fuel consumption of compressor $e \in E_C$
γ_e	constant for calculation of fuel gas consumption of compressor $e \in E_C$
N^{max}_e	maximal power of compressor $e \in E_C$
N^{min}_e	minimal power of compressor $e \in E_C$
q^{min}_v	minimal flow in node $v \in V$
q^{max}_v	maximal flow in node $v \in V$
p^{min}_v	minimal pressure in a node $v \in V$
p^{max}_v	maximal pressure in a node $v \in V$
$p^i_{e,u}$	discretisation values for pressure in the node at beginning of segment $e \in E_P \cup E_C$
$p^i_{e,v}$	discretisation values for pressure in the node at end of segment $e \in E_P \cup E_C$
q^i_e	discretisation values for flow of segment $e \in E_P \cup E_C$
f^i_e	discretisation values for fuel consumption of segment $e \in E_C$

4.4.2 Inequality System

With the definitions of the last subsection we get the following MILP (see the next page):

$$
\begin{array}{rrcll}
 & \min \sum_{e\in E_C} f_e & & & \\
 & \text{s.t.} & & & \\
(1) & q_e & \leq & q_e^{max} s_e & \forall e \in E_R \\
(2) & q_e & \geq & q_e^{min} s_e & \forall e \in E_R \\
(3) & p_{in}^e - p_{out}^e & \leq & dp_e^{max} s_e + M_e(1-s_e) & \forall e \in E_R \\
(4) & p_{in}^e - p_{out}^e & \geq & dp_e^{min} s_e - M_e(1-s_e) & \forall e \in E_R \\
(5) & q_e & \leq & q_e^{max} s_e & \forall e \in E_V \\
(6) & q_e & \geq & q_e^{min} s_e & \forall e \in E_V \\
(7) & M_e s_e - p_{in}^e + p_{out}^e & \leq & M_e & \forall e \in E_V \\
(8) & M_e s_e + p_{in}^e - p_{out}^e & \leq & M_e & \forall e \in E_V \\
(9) & q_e & \leq & q_e^{max} s_e & \forall e \in E_C \\
(10) & q_e & \geq & q_e^{min} s_e & \forall e \in E_C \\
(11) & s_e + s_{v_e} & = & 1 & \forall e \in E_C \\
(12) & p_{in}^e - p_{out}^e & \geq & dp_{v_e}^{max}(s_{v_e} - 1) & \forall e \in E_C \\
(13) & p_{in}^e - p_{out}^e & \leq & 0 & \forall e \in E_C \\
(14) & f_e & \leq & f_e^{max} s_e & \forall e \in E_C \\
(15) & f_e & = & \gamma_e N_e & \forall e \in E_C \\
(16) & N_e & \leq & N_e^{max} s_e & \forall e \in E_C \\
(17) & N_e & \geq & N_e^{min} s_e & \forall e \in E_C \\
(18) & q_e^{min} & \leq & q_e & \forall e \in E_A \\
(19) & q_e & \leq & q_e^{max} & \forall e \in E_A \\
(20) & p_{in}^e & = & p_{out}^e & \forall e \in E_A \\
(21) & \sum_{e\in\delta^+(v)} q_e & = & \sum_{e\in\delta^-(v)} q_e & \forall v \in V \\
(22) & q_v^{min} & \leq & q_v & \forall v \in V \\
(23) & q_v & \leq & q_v^{max} & \forall v \in V \\
(24) & p_v^{min} & \leq & p_v & \forall v \in V \\
(25) & p_v & \leq & p_v^{max} & \forall v \in V \\
(26) & q_e^{min} & \leq & q_e & \forall e \in E_P \\
(27) & q_e & \leq & q_e^{max} & \forall e \in E_P \\
(28) & \sum_{i\in\Lambda_e} \lambda_e^i & = & 1 & \forall e \in E_P \\
(29) & \sum_{j\in Y_e} y_e^j & = & 1 & \forall e \in E_P \\
(30) & y_e^j & \leq & \sum_{k\in N_e^j} \lambda_e^k & \forall e \in E_P,\ \forall j \in Y_e \\
(31) & p_u & = & \sum_{i\in\Lambda_e} p_{e,u}^i\ \lambda_e^i & \forall e = (u,v) \in E_P \\
(32) & p_v & = & \sum_{i\in\Lambda_e} p_{e,v}^i\ \lambda_e^i & \forall e = (u,v) \in E_P \\
(33) & q_e & = & \sum_{i\in\Lambda_e} q_e^i\ \lambda_e^i & \forall e \in E_P \\
(34) & p_{in}^e - p_{out}^e & \geq & 0 & \forall e \in E_P \\
(35) & \sum_{i\in\Lambda_e} \lambda_e^i & = & s_e & \forall e \in E_C \\
(36) & \sum_{j\in Y_e} y_e^j & = & s_e & \forall e \in E_C \\
(37) & y_e^j & \leq & \sum_{k\in N_e^j} \lambda_e^k & \forall e \in E_C,\ \forall j \in Y_e \\
(38) & p_u & = & \sum_{i\in\Lambda_e} p_{e,u}^i\ \lambda_e^i + p_u^h & \forall e = (u,v) \in E_C \\
(39) & p_v & = & \sum_{i\in\Lambda_e} p_{e,v}^i\ \lambda_e^i + p_v^h & \forall e = (u,v) \in E_C \\
(40) & q_e & = & \sum_{i\in\Lambda_e} q_e^i\ \lambda_e^i & \forall e \in E_C \\
(41) & f_e & = & \sum_{i\in\Lambda_e} f_e^i\ \lambda_e^i & \forall e \in E_C \\
(42) & p_u^{max}(s_e - 1) + p_u^h & \leq & 0 & \forall e = (u,v) \in E_C \\
(43) & p_v^{max}(s_e - 1) + p_v^h & \leq & 0 & \forall e = (u,v) \in E_C \\
 & & & & \\
 & \lambda_e^i & \geq & 0 & \forall e \in E_P\ \forall i \in \Lambda_e \\
 & \lambda_e^i & \geq & 0 & \forall e \in E_C\ \forall i \in \Lambda_e \\
 & y_e^j & \in & \{0,1\} & \forall e \in E_P\ \forall j \in Y_e \\
 & y_e^j & \in & \{0,1\} & \forall e \in E_C\ \forall j \in Y_e \\
 & s_e & \in & \{0,1\} & \forall e \in E_R \\
 & s_e & \in & \{0,1\} & \forall e \in E_V \\
 & s_e & \in & \{0,1\} & \forall e \in E_C \\
 & E & = & E_P \,\dot\cup\, E_C \,\dot\cup\, E_V \,\dot\cup\, E_R \,\dot\cup\, E_A &
\end{array}
$$

We remark that generally all variables are nonnegative (with respect to the well known exceptions).

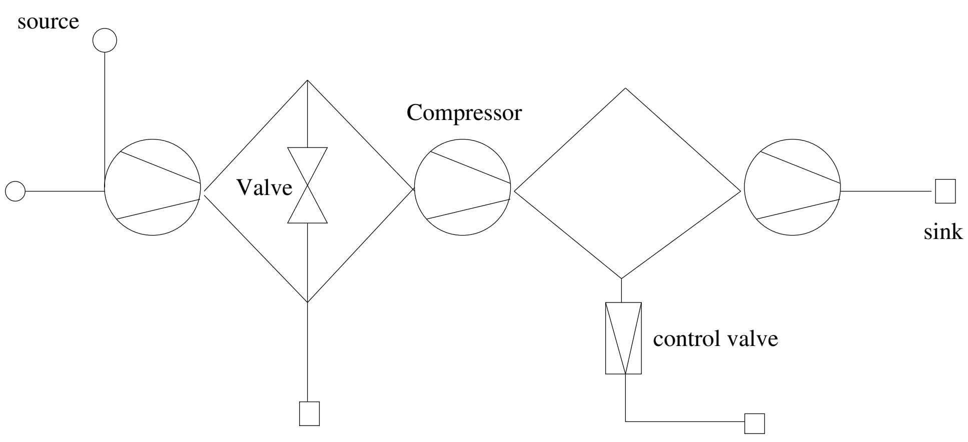

Figure 4.12: A small gas network

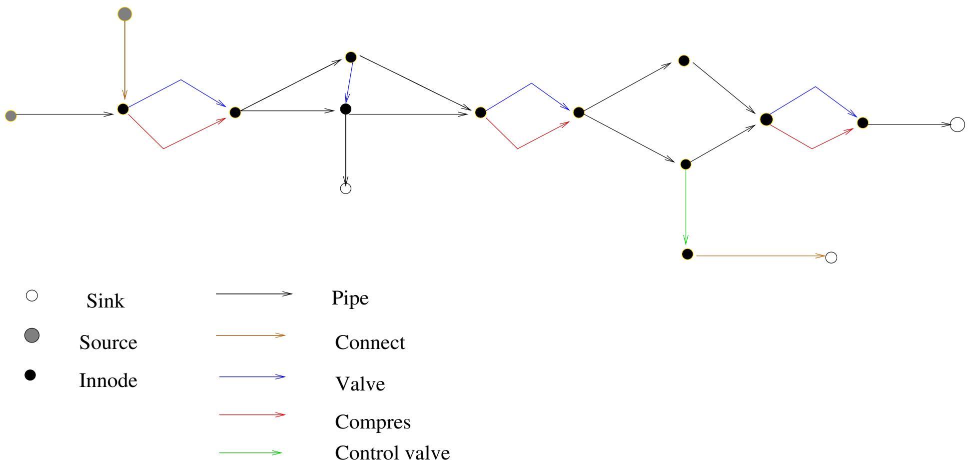

Figure 4.13: Graph of Figure 4.12

4.5 Computational Results

We have implemented the described MILP and have tested it for the small gas network shown in Figure 4.12 which consists of eleven pipes, one connection, four valves, one control valve, three compressors, two sources, three sinks and eleven innodes (the intersection points of segments, see Chapter 4). Note that bypass valves for compressors are not drawn in the figure. The graph $G = (V, E)$ of the test model shown in Figure 4.12 is given in Figure 4.13. The following table shows our first experiences of the computational situation when solving the defined MILP with a standard solver (CPLEX, [25], [26]):

compressors			pipes		Solution			
$p_{in,C}$	$p_{out,C}$	q_C	$p_{in,P}$	q_P	Var.	Ineq.	Opt value	time(sec)
2	2	2	4	10	567	253	7.97	2.85
2	2	2	4*	10*	666	297	7.06	8.14
3	3	7	4	10	801	322	10.79	5.82
3	3	7	4*	10*	900	366	9.49	25.21
4	4	10	4	10	1263	493	10.56	40.83
4	4	10	4*	10*	1362	537	9.37	295.99

We remark that in the table above $*$ means that we have implemented an additional refinement step. In this refinement step we have introduced one additional grid point at the optimum solution of the MILP

and solved the problem again. The values $p_{in,C}$, $p_{out,C}$, q_C, $p_{in,P}$, q_P define the number of intervals we divide the domain of the pressure loss function or the fuel gas consumption function according to the variables of these functions. The column with the entry $Var.$ gives the numbers of variables of the model and the column with the entry $Ineq.$ the number of inequalities of the model.
We remark that CPLEX did not calculate the model for finer triangulations in acceptable time even for this small test network. From the test calculations we also see that the calculation time is distinct dependend on the accuracy of the model. This implies that approximating this nonlinear problem by a MILP we have to find a compromise between exactness and calculation time. The ideas of Chapter 5 will be helpful in order to fasten the calculations.

4.6 Preprocessing

Here we only want to give a short note that there are several possibilities in order to formulate the polyhedron that describes the conditions of a (bypass) valve. In the following lemma P^1 and P^2 are the generalisations of two formulations we tried out in our model. Polyhedron P^2 in Lemma 15 tightens the formulation of P^1 (compare inequalities $5, 12, 7, 8, 6$ in 4.4.2, the last inequality with $q_e^{min} = 0$ and additional bounds) and so it is also possible to implement the polyhedron in this way.

Lemma 15 *Let $x_1, x_3, x_4 \in \mathbb{R}_+$ and $x_2 \in \{0,1\}$. Let a, b, c, d be positive real numbers with $c - b > 0, c - b < d, c - b < a < d$ and*

$$P^1 = \{x_1 \quad \begin{array}{rrrrl} & -ax_2 & & & \leq 0 \\ & -(c-b)x_2 & +x_3 & -x_4 & \geq -(c-b) \\ & dx_2 & -x_3 & +x_4 & \leq d \\ & dx_2 & +x_3 & -x_4 & \leq d \\ x_1 & & & & \geq 0 \\ x_1 & & & & \leq a \\ & & x_3 & & \geq b \\ & & x_3 & & \leq c \\ & & & x_4 & \geq b \\ & & & x_4 & \leq c\}, \end{array}$$

and

$$P^2 = \{x_1 \quad \begin{array}{rrrrl} & -ax_2 & & & \leq 0 \\ & (c-b)x_2 & -x_3 & +x_4 & \leq c-b \\ & (c-b)x_2 & +x_3 & -x_4 & \leq c-b \\ x_1 & & & & \geq 0 \\ & & x_3 & & \geq b \\ & & x_3 & & \leq c \\ & & & x_4 & \geq b \\ & & & x_4 & \leq c\}. \end{array}$$

Then $P^1 = P^2$. If in P^1 we omit the inequality $-(c-b)x_2 + x_3 - x_4 \geq -(c-b)$ the equality $P^1 = P^2$ also holds.

Proof. Obvious. □

Polyhedron P^1 in Lemma 16 is the generalisation of the polyhedron that describes the conditions of a control valve (compare inequalities $1, 3, 4, 2$ in 4.4.2, the last inequality with $q_e^{min} = 0$ and additional bounds). Polyhedron P^2 again gives us an alternative formulation of P^1.

Lemma 16 *Let $x_1, x_3, x_4 \in \mathbb{R}_+$ and $x_2 \in \{0,1\}$. Let a,b,c,d,e,f be positive real numbers with $0 < c < d < e < f < a < b$ and*

$$P^1 = \begin{array}{lllll} \{x_1 & -ax_2 & & & \leq 0 \\ & (b-d)x_2 & +x_3 & -x_4 & \leq b \\ & (b+c)x_2 & -x_3 & +x_4 & \leq b \\ x_1 & & & & \geq 0 \\ x_1 & & & & \leq a \\ & & x_3 & & \geq e \\ & & x_3 & & \leq f \\ & & & x_4 & \geq e \\ & & & x_4 & \leq f\}, \end{array}$$

and

$$P^2 = \begin{array}{lllll} \{x_1 & -ax_2 & & & \leq 0 \\ & (f-e-d)x_2 & +x_3 & -x_4 & \leq f-e \\ & (f-e+c)x_2 & -x_3 & +x_4 & \leq f-e \\ & cx_2 & -x_3 & & \leq -e \\ & cx_2 & & +x_4 & \leq f \\ x_1 & & & & \geq 0 \\ & & x_3 & & \leq f \\ & & & x_4 & \geq e\}. \end{array}$$

Then $P^1 = P^2$.

Proof. Let $x_2 = 0$:
We only have to remember that holds $x_3 \leq f$ and $x_4 \geq e$ and we get $x_3 - x_4 \leq f - e < b$.
Let $x_2 = 1$:
Here with $x_4 \geq e$ we get $x_3 \geq x_4 + c \geq e + c$ from the condition $x_3 - x_4 \geq c$ in P^1 and analogously because of $x_3 \leq f$ we calculate $x_4 \leq x_3 - c \leq f - c$.
With an easy inspection of the further inequalities of P^1 and P^2 we conclude $P^1 = P^2$. □

For the other substructures of our model like the Kirchhoff conditions or the conditions of a compressor no such easy tightened formulations could be constructed.

Considering the polyhedron that describes the gas flow preservation law in all nodes we mention that in the preprocessing step linear dependent conditions are removed without changing this subpolyhedron. Also it is well known that a node-arc incidence matrix of the underlying digraph G is totally unimodular, see [32]. This means that the sets of feasible solutions of the underlying network flow problem are integral subpolyhedra.

4.7 Conclusions

We have shown that in the described way we can define a useful MILP for the optimisation of gas networks. But it is clear that the presented first computational results for our small test model show that we cannot act on the assumption that this model is able to solve reasonable gas networks in short times since the solving times for the test model are relatively high. So we now have to search for a better way in order to fasten the solution time of the model. Because of this we now deal with polyhedral studies of the model and with separation algorithms for the underlying polyhedron or sub-polyhedra. When we have understood the model in a better way we will be able to solve bigger problems (how far we will be able to solve realistic problems is vague since later calculations showed that unfortunately all ideas we have developed will not be enough to solve the problem of a real-world gas transmission company). The next step after this will be that we generalise the stationary model to the transient case which means that the flow and pressure values do not longer need to be constant in time. But such a transient model can only be solved in a reasonable time if we succeed in finding good and fast separation algorithms since the binary variables we had to introduce for the linearisation of the pipes and the compressors increase the solution time of such MILP's. We will search for such possibilities in the next chapter but we antcipate that the ideas we deveolped and implemented will not be enough for the solution of big time depending networks.

Chapter 5

Cutting Planes and Separation Algorithms

5.1 Introduction

Since we want to solve the mixed integer program of the TTO-model via a branch-and-cut algorithm we present in this chapter cutting planes which are useful or potentially useful for solving mixed integer programs that arise in the optimisation of gas networks (see the description in Chapter 4). We consider polyhedra that are defining essential parts of the model (important substructures like the interface of several segments) and give a polynomial algorithm for the calculation of the set of vertices of such polyhedra implying that a polynomial separation algorithm for the convex hull of the polyhedra can be developed.
We also point out how this knowledge can be generalised to more complex structures. Finally our preliminary computational results show the benefits when incorporating these cuts into a general mixed integer programming solver. One important part of our mathematical analysis is that we want to get rid of the binary variables that we introduced for the approximation of nonlinear functions. In Chapter 4 we showed the traditional way for the approximation of nonlinear functions via introducing binary variables. Here we develop an extension of 3.7. Our computational results show the benefits of this method.

5.2 The Polyhedron

From Chapter 4 we only need to remember the flow variables $q_e, e \in E$ and the pressure variables $p_v, v \in V$. Also remember that p_{in} describes the gas pressure in the node at the beginning of a segment and p_{out} means the gas pressure in the node at the end of a segment.
The basic idea behind our polyhedral studies is that the pressure at the end of all ingoing segments of a node must be equal the pressure at the beginning of all outgoing segments of the same node.

Now let us shortly describe how the polyhedron under investigation comes upon in the global model. We have already described and modelled the physics of the gas flowing through a gas network. Remember that the pressure drop in pipes can be approximated by

$$p_{out}^2 = p_{in}^2 - \mathrm{ff}\, q\, |q|,$$

where

$$\mathrm{ff} = \mathrm{ff}(p_{out},\, p_{in})$$

is the friction factor. After simplifying the friction factor to a constant we get $p_{out} = p_{out}(p_{in}, q)$, where p_{out} means the pressure at the end of the pipe, p_{in} means the pressure at the beginning of the pipe and q means the gas flow through the pipe.
The well known fuel gas consumption of the compressors analogously is described by a nonlinear function f of the form: $f = f(p_{in}, p_{out}, q)$. Here f describes the fuel consumption of the compressor, p_{in}

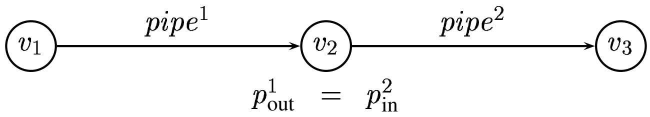

Figure 5.1: Sequence of pipes

the pressure of the gas at the beginning of the compressor, p_{out} the gas pressure which the compressor has to constitute at the endpoint of the compressor and q stands for the gas flow through the compressor. In order to come up with a mixed integer linear program these two nonlinear functions are approximated by suitable triangulations as pointed out in the following demonstrations.

The first substructure of the model we have studied are sequences of pipes. The situation is shown in Figure 5.1.

We have already mentioned one important aspect of the model that the pressure p^1_{out} at the end of the ingoing pipe ($pipe^1$) must be equal the pressure p^2_{in} at the beginning of the outgoing pipe ($pipe^2$). We already know that p^1_{out} is a nonlinear function depending on the flow through the pipe and the pressure at the beginning of the pipe. We approximate the pressure loss in pipes by determing a **triangulation** of the 2-dimensional manifold describing the pressure loss in the pipes. We denote by Λ^{pipe} the set of grid points and by Y^{pipe} the set of triangles. We approximate the $2-$dimensional function $p_{out}(p_{in}, q)$ by linearising it within each triangle. Modelling this piecewise linear approximation results in the following non convex polyhedron:

$$P_\Delta = \Big\{ \tbinom{\lambda^1}{\lambda^2} \in \mathbb{R}^{|\Lambda^1|+|\Lambda^2|} \mid \begin{array}{rcl} \displaystyle\sum_{j\in\Lambda^1} \lambda^1_j & = & 1 \\ \displaystyle\sum_{j\in\Lambda^2} \lambda^2_j & = & 1 \\ \displaystyle\sum_{j\in\Lambda^1} p^1_{\text{out},j}\,\lambda^1_j - \sum_{j\in\Lambda^2} p^2_{\text{in},j}\,\lambda^2_j & = & 0 \\ \lambda^1_j, \quad \lambda^2_j & \geq & 0 \\ \lambda^1,\ \lambda^2 \text{ satisfy the triangle condition} & & \Big\}, \end{array}$$

where the **triangle condition** states that the set of $\lambda-$variables which are strictly positive must belong to grid points of a distinct triangle.
Figure 5.2 describes the situation of the polyhedron P_Δ: The numbers in the left triangulation (for the ingoing pipe) stand for the pressure values $p^1_{out,j}$ at the grid points $j \in \Lambda^1$ and the numbers in the right triangulation (for the outgoing pipe) stand for the pressure values $p^2_{in,i}$ at the grid points $i \in \Lambda^2$. Let us consider a simple example (see Figure 5.3) for a little calculation. Here is $p^1_{out,1} = 10$, $p^1_{out,2} = 8$, $p^1_{out,3} = 4$ and so on, analogously we have for $p^2_{in,1} = p^2_{in,2} = p^2_{in,3} = 10, \ldots$, etc. Consider

$$\lambda^1_1 = \frac{1}{4}, \lambda^1_2 = 0, \lambda^1_3 = 0, \lambda^1_4 = \frac{1}{2}, \lambda^1_5 = \frac{1}{4}, \lambda^1_6 = 0$$

and

$$\lambda^2_1 = \frac{7}{20}, \lambda^2_2 = 0, \lambda^2_3 = 0, \lambda^2_4 = 0, \lambda^2_5 = \frac{13}{20}, \lambda^2_6 = 0.$$

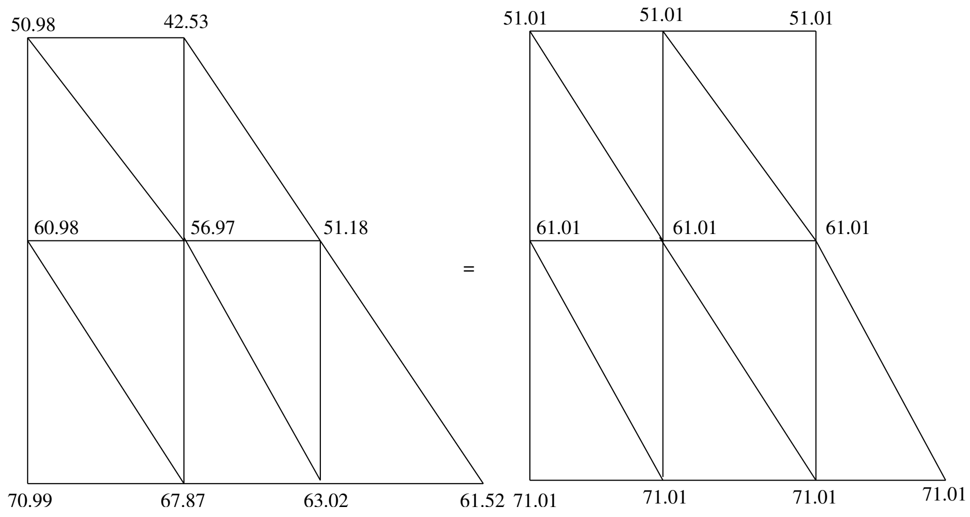

Figure 5.2: Typical triangulation of the pressure loss in a pipe

This setting for the $\lambda-$variables fulfils all conditions, especially the triangle condition.
But if we take

$$\lambda_1^1 = \frac{1}{4}, \lambda_2^1 = 0, \lambda_3^1 = 0, \lambda_4^1 = \frac{1}{2}, \lambda_5^1 = \frac{1}{4}, \lambda_6^1 = 0$$

and

$$\lambda_1^2 = \frac{7}{20}, \lambda_2^2 = 0, \lambda_3^2 = 0, \lambda_4^2 = 0, \lambda_5^2 = 0, \lambda_6^2 = \frac{13}{20},$$

we see that the triangle condition is not satisfied since the nonzero variables λ_1^2 and λ_6^2 belong to two **different** triangles of the triangulation and so this point is no element of $P_\triangle$. In the following we want to generalise our ideas (remember how we approximated the fuel gas consumption of a compressor). Clearly the sequence of two pipes is of course only the simplest case we are faced with. We want to examine the problem more general, where we consider the case that we have an arbitrary number in of ingoing segments and an arbitrary number *out* of outgoing segments. A segment can now be either a pipe or a compressor (but we can as a matter of principle take valves or control valves as segments as we will see later). For every in- and outgoing segment we determine a certain triangulation. In the general case these triangulations do not need to consist only of such regular triangles as in Figure 5.2. The structure can be much more complicated. Perhaps we can not only consider triangles but also squares, pentagons, sexangles, heptagons and so on. Even arbitrary mixtures in the triangulations are possible although this is not interesting for a concrete gas network. And we do not only describe the pressure in the segments but also the gas flow in the segments. Very important for the general formulation is the first law of Kirchhoff which means that the sum of the ingoing gas flows must be equal to the sum of the outgoing gas flows. So in principle (see e.g. [38], [39]) we get the situation which is shown in Figure 5.4.

The requirements of the triangle conditions of $P_\triangle$ are now generalised in the following way:
The triangle conditions mean that for every segment only special combinations of $\lambda-$variables are allowed. For $P_\triangle$ this means that only $\lambda-$variables may be positive that belong to exactly one certain triangle. In the general case only the elements of special sets of $\lambda-$variables may not vanish (Indeed: the reader can recognise that our conditions are a generalised form of Special Ordered Sets (SOS) of type 2, see e.g. [4]). Before going into the details we need to fix some notation.

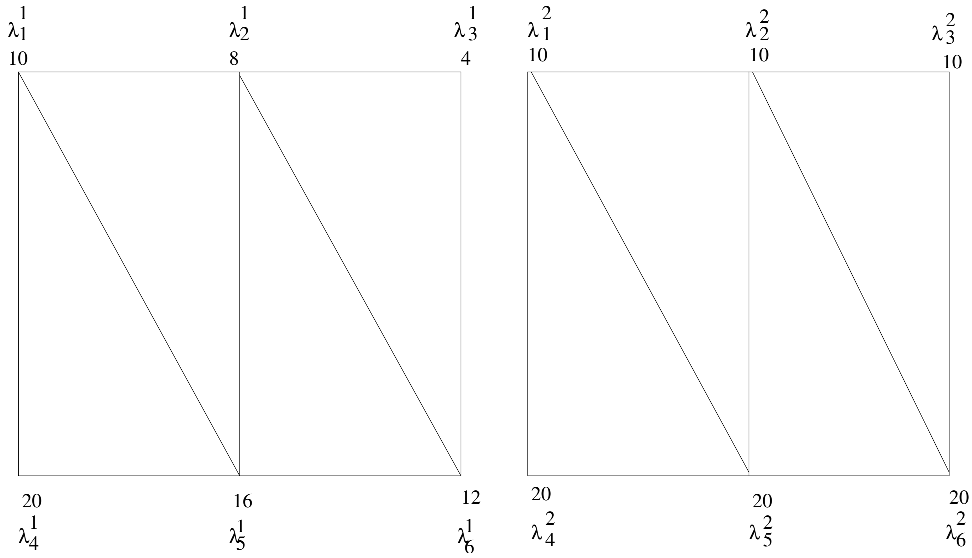

Figure 5.3: An easy example for a triangulation

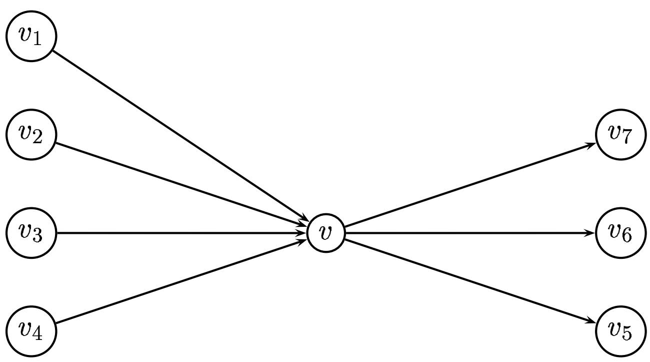

Figure 5.4: Ingoing and outgoing segments in a node

Notation

In this section we give some mathematical notation which is necessary in order to formalise and generalise the above approach.
Let $in \in \mathbb{N}$ be the number of ingoing segments and $out \in \mathbb{N}$ be the number of outgoing segments. A segment may be a pipe or a compressor but also the other types of segments, i.e., valves, control valves and connections (short pipes without pressure loss) can be included in this model. In the mathematical formulation of the model we are no longer bounded to the physical background of the model.
We define a set N^i of grid points for every segment $i \in \{1, 2, \ldots, in + out\}$. W.l.o.g. we assume the ingoing segments to be $1, 2, \ldots, in$ and the outgoing segments to be $in + 1, in + 2, \ldots, in + out$. Furthermore we assume:

$$N^i \cap N^j = \emptyset \qquad \forall i \neq j.$$

We denote by

$$\mathcal{N} = \{N^i \mid i = 1, 2, \ldots, in + out\}$$

the **list** of sets of grid points.
$\mathbb{R}^{N^i}$ denotes the $|N^i|-$dimensional vector space where the components are indexed by N^i and $\mathbb{R}^{\mathcal{N}}$ is defined as:

$$\mathbb{R}^{\mathcal{N}} = \bigotimes_{i=1}^{in+out} \mathbb{R}^{N^i}.$$

We remark that for $\lambda \in \mathbb{R}^{\mathcal{N}}$ we write

$$\lambda = \begin{pmatrix} \lambda^1 \\ \lambda^2 \\ \vdots \\ \lambda^{in+out} \end{pmatrix}$$

with $\lambda^i \in \mathbb{R}^{N^i}$ for all $i \in \{1, 2, \ldots, in + out\}$.
For a list $\mathcal{S}$ of sets of the form

$$\mathcal{S} = \{S^1, S^2, \ldots, S^{in+out}\}$$

we say for some index $j \in \bigcup_{i=1}^{in+out} N^i$:

$$j \in \mathcal{S} \quad \text{iff} \quad \exists\, i \in \{1, 2, \ldots, in + out\} \quad \text{with} \quad j \in S^i.$$

We define

$$\mathcal{S} \subseteq \mathcal{N} \quad \Leftrightarrow \quad \emptyset \neq S^i \subseteq N^i \quad \forall i \in \{1, 2, \ldots, in + out\}.$$

The cardinality of $\mathcal{S}$ is set to

$$|\mathcal{S}| = \sum_{i=1}^{in+out} |S^i|.$$

The characteristic vector of $\mathcal{S}$, which we denote by $\mathcal{X}^{\mathcal{S}} \in \mathbb{R}^{\mathcal{N}}$, is obtained by setting

$$\mathcal{X}_j^{\mathcal{S}} = \begin{cases} 1 & \text{, if } j \in \mathcal{S} \\ 0 & \text{, else.} \end{cases}$$

For each $N^i, i \in \{1, 2, \ldots, in + out\}$ we define n_i subsets $N_k^i, k \in \{1, 2, \ldots, n_i\}$ with

$$N^i = \bigcup_{k=1}^{n_i} N_k^i \quad \text{and} \quad |N_k^i| \geq 2.$$

As an example: In Figure 5.2 holds $n_1 = 8$, $n_2 = 9$ and $|N_k^i| = 3$ for all i, k. We say that a vector $\lambda \in \mathbb{R}^{\mathcal{N}}, \lambda \geq 0$ satisfies the **set condition** (which is the generalisation of the triangle condition in the case of polyhedron P_Δ on page 57) if for all $i = 1, 2, \ldots, in + out$ there exists one $k_i \in \{1, 2, \ldots, n_i\}$ such that

$$\{j \in N^i | \lambda_j^i > 0\} \subseteq N_{k_i}^i.$$

In other words, the set condition holds if for all in- and outgoing segments the non vanishing $\lambda-$variables belong to exactly one of the subsets N_k^i. We say that $\mathcal{S}$ fulfils the set condition if $\mathcal{X}^{\mathcal{S}}$ fulfils the set condition.

Now we define a polyhedron P by

$$P = \{\lambda \in \mathbb{R}^{\mathcal{N}} | A\lambda = b, \lambda \geq 0\},$$

where $A \in \mathbb{R}^{M \times \mathcal{N}}, b \in \mathbb{R}^M$ for some finite set M. We will say something about the cardinality of the set M in the next subsection when we discuss the special structure of the matrix A.
Let us remember that we have already defined for $A \in \mathbb{R}^{M \times \mathcal{N}}$ with

$$A = (a_{ij})_{\substack{\text{i=1,...,m} \\ \text{j=1,...,n}}}$$

and a subset $J \subseteq \{1, 2, \ldots, n\}$ the matrix A_J by

$$A_J = (a_{ij})_{\substack{i \in M \\ j \in J}}$$

Here $m = |M|$ and $n = |\mathcal{N}|$. Analogously we use for $\lambda \in \mathbb{R}^{\mathcal{N}}$ and a subset $J \subseteq \{1, 2, \ldots, n\}$

$$\lambda_J = (\lambda_j)_{j \in J}.$$

For $x \in \mathbb{R}^{\mathcal{S}}$ with $\mathcal{S} \subseteq \mathcal{N}$ we define the **zero-extension** $x^0(\mathcal{S}) \in \mathbb{R}^{\mathcal{N}}$ of x by

$$x^0(\mathcal{S}) = \begin{cases} x_i & \text{, if } i \in \mathcal{S}, \\ 0 & \text{, if } i \in \mathcal{N} \setminus \mathcal{S}. \end{cases}$$

We remark that in the definition of the zero-extension we define $\mathcal{N} \setminus \mathcal{S}$ by

$$\mathcal{N} \setminus \mathcal{S} :\Leftrightarrow N^i \setminus S^i \qquad \forall i \in \{1, 2, \ldots, in + out\}.$$

At the end of this section we define the meaning of $\mathcal{S} \subseteq \bar{\mathcal{S}}$ for two lists $\mathcal{S}$ and $\bar{\mathcal{S}}$:

Let

$$\mathcal{S} = \{S^1, S^2, \ldots, S^{in+out}\}$$

and

$$\bar{\mathcal{S}} = \{\bar{S}^1, \bar{S}^2, \ldots, \bar{S}^{in+out}\}$$

two sets (in the sense of this section). We define:

$$\begin{aligned} \mathcal{S} \subseteq \bar{\mathcal{S}} \quad :\Leftrightarrow \quad & S^1 \subseteq \bar{S}^1 \\ & S^2 \subseteq \bar{S}^2 \\ & \vdots \\ & S^{in+out} \subseteq \bar{S}^{in+out}. \end{aligned}$$

5.3 The Problem

Using the above notation we are now ready to introduce the polyhedron we are going to investigate in this chapter. Remember that we want to model the situation that there are in ingoing and out outgoing segments at some node in the gas network. So we consider a polyhedron P with the following structure:

$$P = \{\lambda \in \mathbb{R}^{\mathcal{N}} | A\lambda = b, \lambda \geq 0 \,, \lambda \text{ satisfies the set conditions}\}.$$

We remark that from our introductory examples it is easy to see that this polyhedron in general is **not** convex.

The special form of the matrix A and the vector b for the general case of polyhedron P becomes

$$A = \begin{pmatrix}
(\mathbb{1}^1)^T &&&&&&& \\
& (\mathbb{1}^2)^T &&&&&& \\
&& \ddots &&&&& \\
&&& (\mathbb{1}^{\mathsf{in}})^T &&&& \\
&&&& (\mathbb{1}^{\mathsf{in}+1})^T &&& \\
&&&&& (\mathbb{1}^{\mathsf{in}+2})^T && \\
&&&&&& \ddots & \\
&&&&&&& (\mathbb{1}^{\mathsf{in}+\mathsf{out}})^T \\
(p^1)^T &&&& -(p^{in+1})^T &&& \\
(p^1)^T &&&&& -(p^{in+2})^T && \\
\vdots &&&&&& \ddots & \\
(p^1)^T &&&&&&& -(p^{in+out})^T \\
& (p^2)^T &&& -(p^{in+1})^T &&& \\
& (p^2)^T &&&& -(p^{in+2})^T && \\
& \vdots &&&&& \ddots & \\
& (p^2)^T &&&&&& -(p^{in+out})^T \\
&& \ddots && \vdots & \vdots & \vdots & \vdots \\
&&& (p^{in})^T & -(p^{in+1})^T &&& \\
&&& (p^{in})^T && -(p^{in+2})^T && \\
&&& \vdots &&& \ddots & \\
&&& (p^{in})^T &&&& -(p^{in+out})^T \\
(q^1)^T & (q^2)^T & \dots & (q^i n)^T & -(q^{in+1})^T & -(q^{in+2})^T & \dots & -(q^{in+out})^T
\end{pmatrix}$$

The entries $p^i \in \mathbb{R}_+^{N^i}, i \in \{1, 2, \ldots, in + out\}$ are vectors describing the pressure at the end of the ingoing and at the beginning of the outgoing segments. Analogously the vectors $q^i \in \mathbb{R}_+^{N^i}, i \in \{1, 2, \ldots, in + out\}$ describe the gas flow in the in- and outgoing segments. These vectors are used to formulate the mentioned first law of Kirchhoff.

The form of the vector b is

$$b = \begin{pmatrix} \mathbb{1}_{\mathsf{in}+\mathsf{out}} \\ 0_{in \cdot out+1} \end{pmatrix}.$$

$\mathbb{1}^{\mathsf{i}} \in \mathbb{R}^{\mathsf{N}^{\mathsf{i}}}$ denotes the vector of all ones. In addition $\mathbb{1}_{\mathsf{m}}$ denotes the vector of all ones in $\mathbb{R}^m$ and 0_m the zero vector in $\mathbb{R}^m$. Let us shortly describe the structure of the matrix A more detailed:

The first in rows describe the sum of the $\lambda-$variables of each ingoing segment. Analogously the next out rows describe the sum of the $\lambda-$variables of each outgoing segment. All sums must be one. In each node there must be a certain pressure. So the rows $in + out + 1$ up to $in + 2out$ describe that the pressure at the end of the first segment must be equal the pressure at the beginning of the outgoing segments. The rows $in + 2out + 1$ up to $(in + 1)(out + 1) - 1$ describe the same situation for the

other ingoing segments combined with the outgoing segments. The last row describes the gas flow in the distinct segments. The gas flow in the outgoing segments is multiplied by -1 because the sum of the gas flows of the ingoing segments must be equal the sum of the gas flows of the outgoing segments. It is easy to see that the matrix A and the vectors λ and b are generalisations of the first discussed situation of one ingoing and one outgoing segment.
As a side remark we want to mention that there are some additional types of segments in a gas network, for example valves, control valves and connections without pressure loss or fuel gas consumption (i.e., there no nonlinear function has to be linearised). In the situation that such an additional segment is an essential part of a subsystem of the gas network also these types of segments can be modelled. Here the vectors for the pressure p or the gas flow q reduce to vectors that are elements from $\mathbb{R}^1$ (In this case the set of grid points for such a segment consists only of one element. Here it is very important to know that it is our aim that we want to cut off LP-solutions, so we can set for these types of segments the pressure and flow values that are calculated in the last iteration. This solution then can be cut off.) because such a segment can in every LP-iteration be interpreted with constant pressure and constant flow and so can be modelled via one single $\lambda-$variable which then has to be one. So the generality of the model is ensured.
When we do not want to include the first law of Kirchhoff, i.e., the gas flow preservation equation in this model, we forget about the last line in $A\lambda = b$. The rang of the Matrix A reduces by one in this case.
We also remark that $|M| = in + (in+1)out + 1 = (in+1)(out+1)$ holds.
For the following considerations it is important to mention that

$$P \subseteq \ [0,1]^{\mathcal{N}}$$

holds, which is easy to see since $\lambda \geq 0$ and because of the definition of the first $in + out$ rows of matrix A and vector b. So the polyhedron P is bounded and we get a non-convex polytop.

5.3.1 The Vertices of the Polyhedron

Let us introduce the idea of calculating the vertices of the polyhedron before we describe the general situation formally in the case of the polyhedron P_Δ: If we want to find a vertex we take one triangle from the triangulation of the ingoing pipe ($pipe^1$) and one triangle from the triangulation of the outgoing pipe ($pipe^2$). To this end we choose some $\lambda-$variables from the selected triangles. Due to the triangle condition the non vanishing $\lambda-$variables at a vertex of P_Δ must belong to exactly one triangle of $pipe^1$ and one triangle of $pipe^2$. Concentrating on two triangles we investigate the extreme points for the selected $\lambda-$variables that fulfil the remaining properties of P_Δ, i.e., if the sum of the selected $\lambda-$variables of $pipe^1$ and the sum of the selected $\lambda-$variables of $pipe^2$ are equal 1, if the pressure equation is fulfilled and of course all $\lambda-$variables we have selected must be nonnegative. We will show that this results in a vertex. By repeating this procedure for all possible selections of $\lambda-$variables we will see that we obtain all vertices of P_Δ. Clearly, if the maximum of the pressure values of $pipe^1$ is lower than the minimum of the pressure values of $pipe^2$ or the minimum of the pressure values of $pipe^1$ is greater than the maximum of the pressure values of $pipe^2$ for a special selection of triangles we do not get a vertex.

Now we give the formal algorithm how the vertices of the polyhedron P can be calculated: Let us begin with the following definition ($rg(A)$ denotes the rang of matrix A):

Definition 17 *We say a subset $\mathcal{S} \subseteq \mathcal{N}$ is* <u>*feasible*</u> *if*

- $|\mathcal{S}| \leq rg(A)$.
- *$\mathcal{S}$ satisfies the set condition.*

Algorithm 18

1. *Set $L = \emptyset$ (the list of all vertices of P).*

$$A = \begin{pmatrix}
(\mathbb{1}^1)^T & & & & & & & & & \\
& (\mathbb{1}^2)^T & & & & & & & & \\
& & \ddots & & & & & & & \\
& & & \ddots & & & & & & \\
& & & & (\mathbb{1}^{\mathsf{in}})^T & & & & & \\
& & & & & (\mathbb{1}^{\mathsf{in}+1})^T & & & & \\
& & & & & & (\mathbb{1}^{\mathsf{in}+2})^T & & & \\
& & & & & & & \ddots & & \\
& & & & & & & & \ddots & \\
& & & & & & & & & (\mathbb{1}^{\mathsf{in}+\mathsf{out}})^T \\
(p^1)^T & & & & & -(p^{in+1})^T & & & & \\
(p^1)^T & & & & & & -(p^{in+2})^T & & & \\
\vdots & & & & & & & \ddots & & \\
\vdots & & & & & & & & \ddots & \\
(p^1)^T & & & & & & & & & -(p^{in+out})^T \\
(p^1)^T & -(p^2)^T & & & & & & & & \\
& (p^2)^T & -(p^3)^T & & & & & & & \\
& \ddots & \ddots & & & & & & & \\
& & \ddots & \ddots & & & & & & \\
& & & (p^{in-1})^T & -(p^{in})^T & & & & & \\
(q^1)^T & (q^2)^T & \dots & \dots & (q^{in})^T & -(q^{in+1})^T & -(q^{in+2})^T & \dots & \dots & -(q^{in+out})^T
\end{pmatrix}$$

Figure 5.5: Simplified Matrix A

2. *For all feasible subsets $\mathcal{S} \subseteq \mathcal{N}$ do*
    (a) *Solve $A_{\mathcal{S}}\,\lambda_{\mathcal{S}} = b$.*
    (b) *If the system has a unique solution $\bar{\lambda}_{\mathcal{S}}$ with $\bar{\lambda}_{\mathcal{S}} \geq 0$, add the zero-extension of $\bar{\lambda}_{\mathcal{S}}$ to L.*

In the following we want to prove that this algorithm runs in polynomial time and computes all vertices of P. As a consequence we obtain that P has only polynomially many vertices.
But at first let us make the following

Remark 19 *The matrix A on page 62 can be simplified to the form in Figure 5.5 and the vector b to*

$$b = \begin{pmatrix} \mathbb{1}_{\mathsf{in}+\mathsf{out}} \\ 0_{in+out} \end{pmatrix}.$$

From this we conclude that

$$in + out \leq rg(A) \leq 2(in + out).$$

Before we prove that the algorithm is correct we discuss the following

Lemma 20 *The described algorithm reduced by the postulation of the set condition can principally also be used in order to calculate the vertices of the polyhedron P without the set condition.*

This lemma is a direct consequence of well known results of linear programming, namely that the support of a vertex of a polyhedron $P = \{x \mid Ax = b, x \geq 0\}$ is at most the number of rows of A. When we consider the polyhedron P without the set conditions this polyhedron is completely described by (in-) equalities and thus the above argument applies. The problem in the case of P (with set conditions) is

that we do not know the complete description of the polyhedron in form of equalities or inequalities and thus this simple argument cannot be used.
In our case we formulate the following

Theorem 21 *The above algorithm is correct, i.e., it calculates all vertices of the polyhedron P.*

For the proof of Theorem 21 we formulate

Lemma 22 *Let $\mathcal{S}$ satisfy the set condition (which is fulfiled by a feasible set because of Definition 17). If $A_{\mathcal{S}}\lambda_{\mathcal{S}} = 0_{|\mathcal{N}|}$ has a nontrivial solution then the zero-extension of a positive solution of $A_{\mathcal{S}}\lambda_{\mathcal{S}} = b$ is not a vertex. Clearly, if $\mathcal{S}$ is not feasible then $|\mathcal{S}| > rg(A)$ and it is well known that in this case $A_{\mathcal{S}}\lambda_{\mathcal{S}} = 0_{|\mathcal{N}|}$ must have a nontrivial solution. So we do not get a vertex if $\mathcal{S}$ is not feasible.*

Proof. Let $\bar{\lambda}_{\mathcal{S}}$ be a positive solution of $A_{\mathcal{S}}\lambda_{\mathcal{S}} = b$, i.e., it holds $A_{\mathcal{S}}\bar{\lambda}_{\mathcal{S}} = b$ with $\bar{\lambda}_{\mathcal{S}} > 0_{|\mathcal{S}|}$.
Let $\bar{\lambda}$ be the zero extension of $\bar{\lambda}_{\mathcal{S}}$. We will show that $\bar{\lambda}$ is a nontrivial convex combination of two other points in P (which are elements of an $\epsilon-$environment ($\epsilon > 0$) of $\bar{\lambda}$). This shows that $\bar{\lambda}$ cannot be a vertex. We define for $\mathcal{S}$ a vector $\epsilon \in \mathbb{R}^{\mathcal{S}}$ with $\mathcal{S} \subseteq \mathcal{N}$ as follows:

From $A_{\mathcal{S}}\bar{\lambda}_{\mathcal{S}} = b$ and the condition $A_{\mathcal{S}}(\bar{\lambda}_{\mathcal{S}} + \epsilon) = b$ we get:

$$A_{\mathcal{S}}\epsilon = 0_{|\mathcal{N}|}.$$

Obviously $\bar{\epsilon} = 0_{|\mathcal{S}|}$ is a solution. We know from the assumptions of Lemma 22 that $A_{\mathcal{S}}\epsilon = 0_{|\mathcal{N}|}$ has a nontrivial solution. Because of this we also know that the set of solutions of $A_{\mathcal{S}}\epsilon = 0_{|\mathcal{N}|}$ is a vector space (with nontrivial solutions). Therefore, there exists $\bar{\epsilon} \neq 0_{|\mathcal{S}|}$ such that $A_{\mathcal{S}}(\bar{\lambda}_{\mathcal{S}} + \bar{\epsilon}) = b$, with $\bar{\lambda}_{\mathcal{S}} + \bar{\epsilon} > 0_{|\mathcal{S}|}$ and $\bar{\lambda}_{\mathcal{S}} - \bar{\epsilon} > 0_{|\mathcal{S}|}$.

Now we built the zero-extension $(\bar{\lambda}_{\mathcal{S}} + \bar{\epsilon})^0(\mathcal{S})$ of $\bar{\lambda}_{\mathcal{S}} + \bar{\epsilon}$ and we get $(\bar{\lambda}_{\mathcal{S}} + \bar{\epsilon})^0(\mathcal{S}) \in P$. Observe that for $\mathcal{S}$ all $\lambda-$variables must fulfil the set condition by construction.
Similarly, $A_{\mathcal{S}}(\bar{\lambda}_{\mathcal{S}} - \bar{\epsilon}) = A_{\mathcal{S}}\bar{\lambda}_{\mathcal{S}} - A_{\mathcal{S}}\bar{\epsilon} = A_{\mathcal{S}}\bar{\lambda}_{\mathcal{S}} - 0_{|\mathcal{N}|} = b$ and hence also $A(\bar{\lambda}_{\mathcal{S}} - \bar{\epsilon})^0(\mathcal{S}) = b$. We conclude $(\bar{\lambda}_{\mathcal{S}} - \bar{\epsilon})^0(\mathcal{S}) \in P$.
Finally,

$$\frac{1}{2}(\bar{\lambda}_{\mathcal{S}} + \bar{\epsilon}) + \frac{1}{2}(\bar{\lambda}_{\mathcal{S}} - \bar{\epsilon}) = \bar{\lambda}_{\mathcal{S}}.$$

and

$$\frac{1}{2}(\bar{\lambda}_{\mathcal{S}} + \bar{\epsilon})^0(\mathcal{S}) + \frac{1}{2}(\bar{\lambda}_{\mathcal{S}} - \bar{\epsilon})^0(\mathcal{S}) = (\bar{\lambda}_{\mathcal{S}})^0(\mathcal{S}) = \bar{\lambda}.$$

Since $\bar{\lambda}$ can be written as a convex sum of two other points of P it cannot be a vertex. □

We use the lemma in the following

Proof. We show Theorem 21 in two steps. At first we show that all calculated points are vertices of P and then we show that there cannot exist other vertices of P.

1) The calculated points are vertices of P.

From the first $in + out$ rows of A it is clear that for every segment at least one variable must be greater than zero. We define for a feasible subset $\mathcal{S} \subseteq \mathcal{N}$ and its characteristic vector $\mathcal{X}^{\mathcal{S}}$ the following inequality:

$$(\mathcal{X}^{\mathcal{S}})^T\lambda \leq in + out.$$

From the definition of P we see that this inequality is valid for P since the sum of all $\lambda-$variables of a point in P is always equal to $in + out$.

Let $\bar{\lambda} = \lambda^0(\mathcal{S}) \in P$ be the zero extension of $\lambda_{\mathcal{S}}$ calculated according the algorithm corresponding to $\mathcal{S}$. We show the following:

$$\{\bar{\lambda}\} = P \cap \{(\mathcal{X}^{\mathcal{S}})^T \lambda = in + out\}.$$

Since $\bar{\lambda} \in P$ by construction and $(\mathcal{X}^{\mathcal{S}})^T \bar{\lambda} = in + out$ by definition of $\mathcal{X}^{\mathcal{S}}$, the first inclusion $\{\bar{\lambda}\} \subseteq P \cap \{(\mathcal{X}^{\mathcal{S}})^T \lambda = in + out\}$ is trivial.

Now we show $\{\bar{\lambda}\} \supseteq P \cap \{(\mathcal{X}^{\mathcal{S}})^T \lambda = in + out\}$.
Suppose there exists another point

$$\tilde{\lambda} \in (P \cap \{(\mathcal{X}^{\mathcal{S}})^T \lambda = in + out\}) \setminus \{\bar{\lambda}\}.$$

Observe that $\tilde{\lambda}_i = 0$ for all $i \notin \mathcal{S}$. This implies that $\tilde{\lambda}$ is another solution to $A_{\mathcal{S}} \lambda_{\mathcal{S}} = b$, a contradiction to the construction of $\bar{\lambda}$.

2) There are no other vertices of P.

We have seen in the first part of this proof that the constructed points are indeed vertices of P. From Lemma 22 it is now easy to see that there are no other vertices of P. W.l.o.g. we can restrict ourselves to feasible sets $\mathcal{S}$ that produce a positive solution $\bar{\lambda}_{\mathcal{S}}$ of $A_{\mathcal{S}} \lambda_{\mathcal{S}} = b$ which is not unique. In this case we apply Lemma 22, because in this case $A_{\mathcal{S}} \lambda_{\mathcal{S}} = 0_{|\mathcal{N}|}$ must have a non-trivial solution. Let $\bar{\lambda} \in P$ be the zero-extension of $\bar{\lambda}_{\mathcal{S}}$. Thus $\bar{\lambda} \in P$ cannot be a vertex. □

From the theorem and its proof above we conclude that the non-convex polyhedron P can be written as a union of convex polytopes. This can be understood in the following way: In the case of the polyhedron P_{Δ} every selection of a triangle of the ingoing pipe combined with a selection of a triangle of the outgoing pipe defines a small polyhedron. By the zero-extension we get a polyhedron in the space of all λ-variables. The non-convex polyhedron P_{Δ} can evidently be understood as the union of all polyhedra in the space of all λ-variables that arise from all possible combinations of a triangle of the ingoing pipe and a triangle of the outgoing pipe. It is obvious that this idea can be extended to the general case of polyhedron P.

As a side remark we notice that from this observation we can get an easy algorithm in order to construct valid inequalities for P from valid inequalities of the convex polytopes whose union is P (see [27]).
Let $P = \bigcup_{k=1,\dots,l} P^k$ be the union of $l \in \mathbb{N}$ convex polytopes.

Lemma 23 *Let $(a^k)^T \lambda \leq \alpha^k$ for $k = 1, \dots, l$ be a valid inequality for polyhedron P^k. Then*

$$\sum_{i=1}^{|\mathcal{N}|} min(a_i^1, a_i^2, \dots, a_i^l)\, \lambda_i \leq max(\alpha^1, \alpha^2, \dots, \alpha^l)$$

is valid for P.

Proof. In the case $l = 1$ nothing is to show. In the case $l = 2$ we calculate for all $\lambda \in P^1$:

$$\sum_{i=1}^{|\mathcal{N}|} min(a_i^1, a_i^2)\, \lambda_i \leq \sum_{i=1}^{|\mathcal{N}|} a_i^1 \lambda_i \leq \alpha^1 \leq max(\alpha^1, \alpha^2)$$

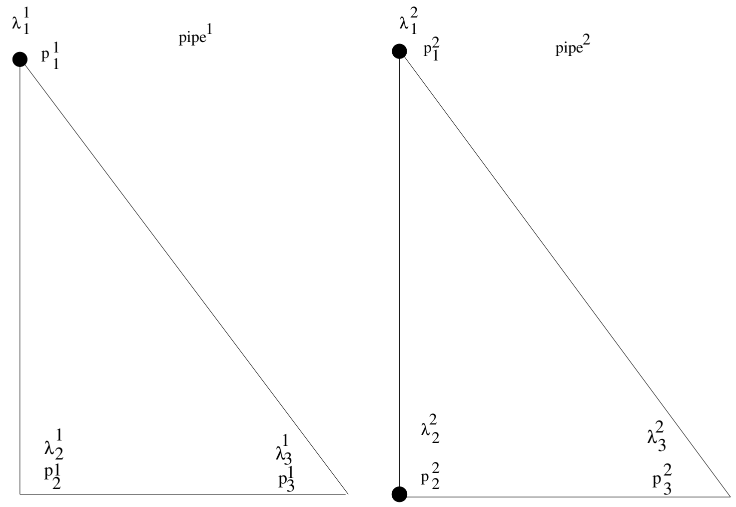

Figure 5.6: Building vertices of the polyhedron P_{Δ}

and analogous for all $\lambda \in P^2$:

$$\sum_{i=1}^{|\mathcal{N}|} min(a_i^1, a_i^2)\,\lambda_i \le \sum_{i=1}^{|\mathcal{N}|} a_i^2 \lambda_i \le \alpha^2 \le max(\alpha^1, \alpha^2).$$

We show the general case by induction over l.
Let $P = \bigcup_{k=1,\ldots,l+1} P^k = \bigcup_{k=1,\ldots,l} P^k \,\cup\, P^{l+1}$. We see at one glance

$$\sum_{i=1}^{|\mathcal{N}|} min(a_i^1, a_i^2, \ldots, a_i^{l+1})\,\lambda_i = \sum_{i=1}^{|\mathcal{N}|} min\Big((a_i^1, a_i^2, \ldots, a_i^l), a_i^{l+1}\Big)\,\lambda_i$$

$$\le max\Big((\alpha^1, \alpha^2, \ldots, \alpha^l), \alpha^{l+1}\Big) = max(\alpha^1, \alpha^2, \ldots, \alpha^{l+1}).$$

□

It is clear that this idea generally only leads to relatively weak inequalities which we can see from [28] (there the inequality is given for $l = 2$) and from the fact that for increasing numbers of polyhedra we can define sequences of valid inequalities such that the constructed inequality can become weaker and weaker.

Let us come back to Algorithm 18 with some examples.

Example 24 *We consider a simple example in order to demonstrate the essential parts of the used notation (not all elements because the notation is much more complex than the idea behind it). Let us consider the following case of polyhedron P_{Δ} (a picture is shown in Figure 5.6. According to the picture holds: $n_1 = n_2 = 1$ with $|N_1^1| = |N_1^2| = 3$): Let the matrix A be:*

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 15 & 10 & 10 & -10 & -10 & -20 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

with

$$p^1 = \begin{pmatrix} 15 \\ 10 \\ 10 \end{pmatrix},$$

$$p^2 = \begin{pmatrix} 10 \\ 10 \\ 20 \end{pmatrix}.$$

Also

$$q^1 = q^2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

The vector b becomes

$$b = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

We have already mentioned that if we do not want to model the gas flow preservation the last row of A can be omitted. We will do this from now on and in all upcoming examples. Because $rg(A) = 3$ we take as a first selection $\mathcal{S}_1 = \{S^1, S^2\}$ with $S^1 = \{1\}$ and $S^2 = \{4, 6\}$. Here $A_{\mathcal{S}_1}$ becomes

$$A_{\mathcal{S}_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 15 & -10 & -20 \end{pmatrix}$$

and according to our algorithm we have to solve:

$$A_{\mathcal{S}_1}\lambda_{\mathcal{S}_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 15 & -10 & -20 \end{pmatrix} \begin{pmatrix} \lambda_1^1 \\ \lambda_1^2 \\ \lambda_3^2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}.$$

We get as a unique (and also nonnegative) solution:

$$\lambda_{\mathcal{S}_1} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}.$$

The zero-extension of $\lambda_{\mathcal{S}_1}$

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{pmatrix}$$

is a vertex of P_Δ.
If we take as a second selection $\mathcal{S}_2 = \{S^1, S^2\}$ with $S^1 = \{2\}$ and $S^2 = \{4, 5\}$ we have to solve:

$$A_{\mathcal{S}_2}\lambda_{\mathcal{S}_2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 10 & -10 & -10 \end{pmatrix} \begin{pmatrix} \lambda_2^1 \\ \lambda_1^2 \\ \lambda_2^2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}.$$

Here $rg(A_{\mathcal{S}_2}) = 2$ and so we know from Theorem 21 that $\mathcal{S}_2$ does not lead to a vertex, since $|\mathcal{S}_2| > 2$. But if we reduce $\mathcal{S}_2$ to $\mathcal{S}_3 = \{S^1, S^2\}$ with $S^1 = \{2\}$ and $S^2 = \{4\}$ we have to solve

$$A_{\mathcal{S}_3}\lambda_{\mathcal{S}_3} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 10 & -10 \end{pmatrix} \begin{pmatrix} \lambda_2^1 \\ \lambda_1^2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix},$$

and we get a unique (and nonnegative) solution

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

which fulfils all demanded properties that we have pointed out in Algorithm 18. Thus the zero-extension of this vector

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

yields a vertex of $P_\triangle$. We will discuss the general case of $P_\triangle$ in the next example in a more detailed way.

We again consider Algorithm 18 which we apply to polyhedron $P_\triangle$:

Example 25 *We analysise a second time the case of one ingoing and one outgoing pipe described on page 57. The polyhedron P defined on page 62 reduces in this case to the polyhedron $P_\triangle$. We now want to describe formally the case that we have described numerically in the last example. The form of the matrix A generally reads:*

$$A = \begin{pmatrix} (\mathbb{1}^1)^T & \\ & (\mathbb{1}^2)^T \\ (p^1)^T & -(p^2)^T \end{pmatrix}.$$

The form of the vectors λ and b are

$$\lambda = \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix},$$

$$b = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}.$$

It is easy to see that $2 \leq rg(A) \leq 3$ and $rg(A) = 2$ if and only if there exist constants c_1, c_2 so that $p^1 = c_1 \mathbb{1}^1$ and $p^2 = c_2 \mathbb{1}^2$. If $c_1 \neq c_2$ the polyhedron is empty. In the case $c_1 = c_2$ we can easily describe the vertices of $P_\triangle$. From Algorithm 18 we know that we have to select feasible sets $\mathcal{S}$ with $|S^1| = |S^2| = 1$. All these possible feasible sets lead to a vertex of $P_\triangle$ in which the two selected $\lambda-$variables in $\mathcal{S}$ get the value 1 (and the not selected $\lambda-$variables in $\mathcal{N} \setminus \mathcal{S}$ are by construction 0).
Now let $rg(A) = 3$. So we can select a feasible set $\mathcal{S}$ with $|S^1| + |S^2| \leq 3$ and there are manifestly the following three possibilities:

- *Select one $\lambda-$variable from N^1 and one $\lambda-$variable from N^2 and try to solve the resulting linear equality system (cf. the case when $rg(A) = 2$ above). Here is $|S^1| = |S^2| = 1$. The principle situation is as in Figure 5.7. If $p_1^1 = p_1^2$ we get a vertex for which $\lambda_1^1 = \lambda_1^2 = 1$ and the other remaining $\lambda-$variables are zero, otherwise we don't get a vertex.*

- *Select the feasible set $\mathcal{S}$ such that one $\lambda-$variable from the ingoing pipe and two $\lambda-$variables from the outgoing pipe are chosen, i.e., formally it holds $|S^1| = 1, |S^2| = 2$. The situation is pointed out in Figure 5.8. If $p_1^2 \leq p_1^1 \leq p_2^2$ or $p_2^2 \leq p_1^1 \leq p_1^2$ (see the figure) and $p_1^2 \neq p_2^2$ we can*

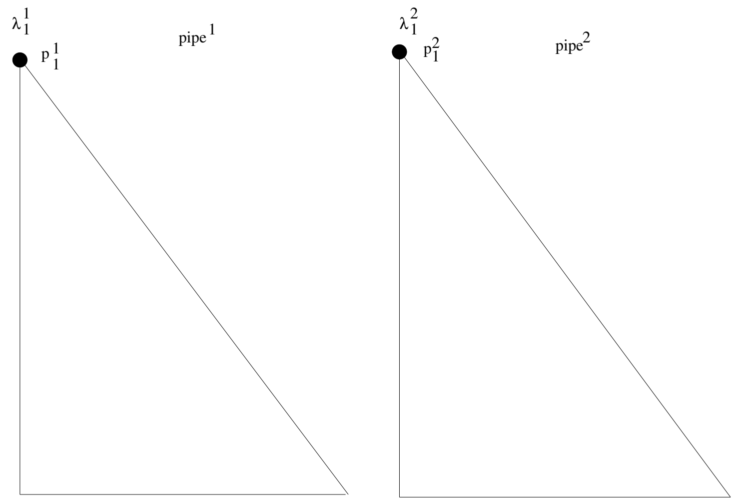

Figure 5.7: Vertices for the polyhedron P_Δ with a selection of two $\lambda-$variables

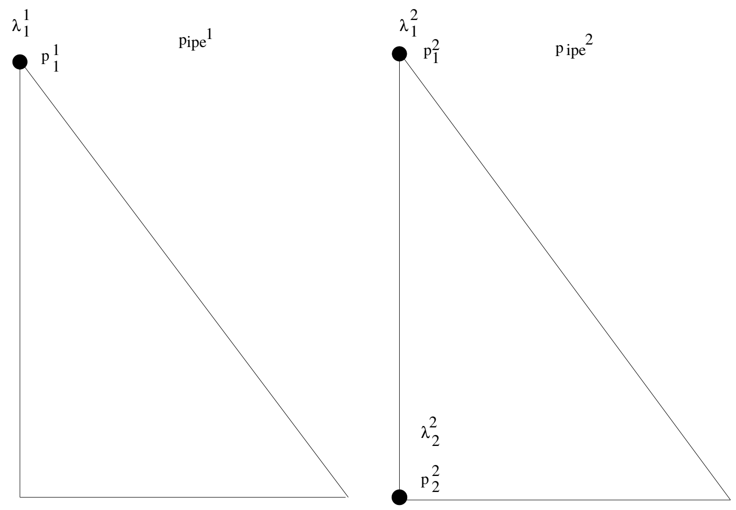

Figure 5.8: Vertices for the polyhedron P_Δ with a selection of three $\lambda-$variables

construct a vertex for which holds (cf. Algorithm 18):

$$\lambda_1^1 = 1 \tag{5.1}$$

$$\lambda_1^2 = \frac{p_1^1 - p_2^2}{p_1^2 - p_2^2} \tag{5.2}$$

$$\lambda_2^2 = \frac{p_1^2 - p_1^1}{p_1^2 - p_2^2} \tag{5.3}$$

The remaining $\lambda-$variables are again set to zero.

Proof. *This is an easy consequence from our algorithm. The reduced linear equation system (see $A_{\mathcal{S}}\lambda_{\mathcal{S}} = b$) reads*

$$\begin{array}{ccccccc} \lambda_1^1 & & & & & = & 1 \\ & & \lambda_1^2 & + & \lambda_2^2 & = & 1 \\ \lambda_1^1 p_1^1 & - & \lambda_1^2 p_1^2 & - & \lambda_2^2 p_2^2 & = & 0 \end{array}$$

Now we see that under our assumptions

$$det \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ p_1^1 & -p_1^2 & -p_2^2 \end{pmatrix} = p_1^2 - p_2^2 \neq 0.$$

And an easy calculation shows that $\lambda_1^1, \lambda_1^2, \lambda_2^2$ are the unique non-negative solution of the above linear equation system $A_{\mathcal{S}}\lambda_{\mathcal{S}} = b$.
Therefore (built again the zero-extension) we have constructed a vertex of $P_\triangle$. □

- *Select a feasible set $\mathcal{S}$ with two selected $\lambda-$variables from N^1 and one $\lambda-$variable from N^2, in which case $|S^1| = 2, |S^2| = 1$.*
  The calculation of the non vanishing values of the vertex is analogous to the previous case.

Only these three types of feasible sets $\mathcal{S}$ possibly result in vertices of $P_\triangle$ because $rg(A) = 3$. We will see an example for a numerical calculation in the next subsection.

One problem while calculating the vertices of polyhedron A is that we have to solve linear equation systems. Because of this we now give a short summary of important cases for which we can give easy formulas for calculating the vertices. In the following ventilations the type of the in- or outgoing segments is of no account.

Remark 26 *In Example 25 we have classified all cases of the polyhedron $P_\triangle$. We remember that we got the situation that we could choose the feasible set $\mathcal{S}$ such that we selected*

(a) one λ-variable for the in- and one λ-variable for the outgoing pipe,

(a) one λ-variable for the in- and two λ-variables for the outgoing pipe,

(a) two λ-variables for the in- and one λ-variable for the outgoing pipe.

We see at one glance that these three cases combined with the formulas for the solution (see (1),(2),(3) in Example 25) also hold if we take another type of segment instead of pipes since Algorithm 18 works for arbitrary sets $\mathcal{N}$.
Let us consider now P again for in ingoing and out outgoing segments. We assume matrix A to have full row rank. If we want to guarantee that we can calculate all components of vertices of P by such

formulas we have given in Example 25 matrix A must have exactly $in + out + 1$ rows, i.e., it has the form

$$A = \begin{pmatrix} (\mathbb{1}^1)^T &&&&&&& \\ & (\mathbb{1}^2)^T &&&&&& \\ && \ddots &&&&& \\ &&& (\mathbb{1}^{\text{in}})^T &&&& \\ &&&& (\mathbb{1}^{\text{in}+1})^T &&& \\ &&&&& (\mathbb{1}^{\text{in}+2})^T && \\ &&&&&& \ddots & \\ &&&&&&& (\mathbb{1}^{\text{in}+\text{out}})^T \\ (v^1)^T & (v^2)^T & \dots & (v^{in})^T & -(v^{in+1})^T & -(v^{in+2})^T & \dots & -(v^{in+out})^T \end{pmatrix}$$

For our problem this means if $in + out \geq 2$ we get the vertices of polyhedron P such that only the first law of Kirchhoff is modelled. In the case $in + out = 2$ we also get the polytop which describes the gas flow preservation of one in- and one outgoing segment and the well known polyhedron P_{Δ}. This simple remark is interesting for us since we can apply the separation algorithm we developed on the basis of Algorithm 18 for a huger class of situations with very easy instruments.
Let us give the complete formulas for the next complexer situation which apparently occurs when we are dealing with the situation $rg(A) = in + out + 2$ which means that A becomes the form

$$A = \begin{pmatrix} (\mathbb{1}^1)^T &&&&&&& \\ & (\mathbb{1}^2)^T &&&&&& \\ && \ddots &&&&& \\ &&& (\mathbb{1}^{\text{in}})^T &&&& \\ &&&& (\mathbb{1}^{\text{in}+1})^T &&& \\ &&&&& (\mathbb{1}^{\text{in}+2})^T && \\ &&&&&& \ddots & \\ &&&&&&& (\mathbb{1}^{\text{in}+\text{out}})^T \\ (v^1)^T & (v^2)^T & \dots & (v^in)^T & -(v^{in+1})^T & -(v^{in+2})^T & \dots & -(v^{in+out})^T \\ (w^1)^T & (w^2)^T & \dots & (w^in)^T & -(w^{in+1})^T & -(w^{in+2})^T & \dots & -(w^{in+out})^T \end{pmatrix}$$

Matrix A again may have full row rank. The most easy and common polyhedron of this type describes the pressure equality in the node at the endpoint of the ingoing segment and the node at the beginning of the outgoing segment **combined** *with the first law of Kirchhoff. The other situations that fulfil $rg(A) = in + out + 2$ are first the sequence of two ingoing and one outgoing segment and second the sequence of one ingoing and two outgoing segments at each case only the pressure equality in the nodes may be regarded.*
Let us shortly give the formulas in the first case we mentioned for P with

$$A = \begin{pmatrix} (\mathbb{1}^1)^T & \\ & (\mathbb{1}^2)^T \\ (p^1)^T & -(p^2)^T \\ (q^1)^T & -(q^2)^T \end{pmatrix}$$

with $rg(A) = 4$ and

$$b = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

We consider the following cases (we remark that we only mention the non-vanishing λ-variables, see Algorithm 18):

1. $\mathcal{S} = 2$ *with* $|S^1| = |S^2| = 1$. *If* $p_i^1 = p_j^2$ **and** $q_i^1 = q_j^2$ *than we get a vertex with* $\lambda_i^1 = \lambda_j^2 = 1$.
2. $\mathcal{S} = 3$ *with* $|S^1| = 1, |S^2| = 2$. *The vertices of this type can be calculated from the vertices of the polyhedron* P_Δ.
   First forget the first law of Kirchhoff. We already know the vertices of the remaining polyhedron P_Δ. *If such a vertex fulfils the first law of Kirchhoff we have found a vertex of* P. *Analogously we now forget the pressure equality condition. We also already know the vertices of the remaining polyhedron. If such a vertex fulfils the pressure equality condition we have found a vertex of* P.
3. $\mathcal{S} = 3$ *with* $|S^1| = 2, |S^2| = 1$. *This case can be managed analogously as case* 2.
4. $\mathcal{S} = 4$ *with* $|S^1| = 2, |S^2| = 2$. *We select the following* λ*-variables:* $\lambda_i^1, \lambda_j^1, \lambda_k^2, \lambda_l^2$*: From* $A_\mathcal{S}\lambda_\mathcal{S} = b$ *we calculate*
$$\lambda_i^1 = \frac{(p_l^2 - p_j^1)(q_l^2 - q_k^2) - (p_l^2 - p_k^2)(q_l^2 - q_j^1)}{(p_i^1 - p_k^1)(q_l^2 - q_k^2) - (q_i^1 - q_j^1)(p_l^2 - p_k^2)}$$
$$\lambda_j^1 = 1 - \lambda_i^1$$
$$\lambda_k^2 = \frac{(q_j^1 - q_i^1)(p_l^2 - p_j^1) + (p_i^1 - p_j^1)(q_l^2 - q_j^1)}{(p_i^1 - p_j^1)(q_l^2 - q_k^2) - (q_i^1 - q_j^1)(p_l^2 - p_k^2)}$$
$$\lambda_l^2 = 1 - \lambda_k^2$$
   If this solution fulfils $2(b)$ *in Algorithm 18 we have found a new vertex (the uniqueness is exactly the case if* $(p_i^1 - p_j^1)(q_l^2 - q_k^2) - (q_i^1 - q_j^1)(p_l^2 - p_k^2) \neq 0$*).*
5. $\mathcal{S} = 4$ *with* $|S^1| = 1, |S^2| = 3$. *We select the following* λ*-variables:* $\lambda_i^1, \lambda_j^2, \lambda_k^2, \lambda_l^2$*: From* $A_\mathcal{S}\lambda_\mathcal{S} = b$ *we calculate*
$$\lambda_i^1 = 1$$
$$\lambda_j^2 = 1 - \lambda_k^2 - \lambda_l^2$$
$$\lambda_k^2 = \frac{(p_i^1 - p_j^2)(q_l^2 - q_j^2) - (p_l^2 - p_j^2)(q_i^1 - q_j^2)}{(p_k^2 - p_j^2)(q_l^2 - q_j^2) - (q_k^2 - q_j^2)(p_l^2 - p_j^2)}$$
$$\lambda_l^2 = \frac{(q_j^2 - q_k^2)(p_i^1 - p_j^2) - (p_k^2 - p_j^2)(q_i^1 - q_j^2)}{(p_k^2 - p_j^2)(q_l^2 - q_j^2) - (q_k^2 - q_j^2)(p_l^2 - p_j^2)}$$
   If this solution fulfils $2(b)$ *in Algorithm 18 we have found a new vertex (the uniqueness is exactly the case if* $(p_k^2 - p_j^2)(q_l^2 - q_j^2) - (q_k^2 - q_j^2)(p_l^2 - p_j^2) \neq 0$*).*
6. $\mathcal{S} = 4$ *with* $|S^1| = 3, |S^2| = 1$. *Of course this case is principle the same as case* 5.

In the other two cases the situation is analogous but these are of minor interest.
Complexer situations are perfunctorily for us (and the calculation of the vertices becomes much more difficult) and so we can implement Algorithm 18 without using a general Gaussian algorithm.

We have already mentioned that the vertices of course can be calculated with the same formulas in the case that one or more of the segments are (switched on) compressors or pipes. So w.l.o.g. we sometimes just restricted us to the case of pipes. □

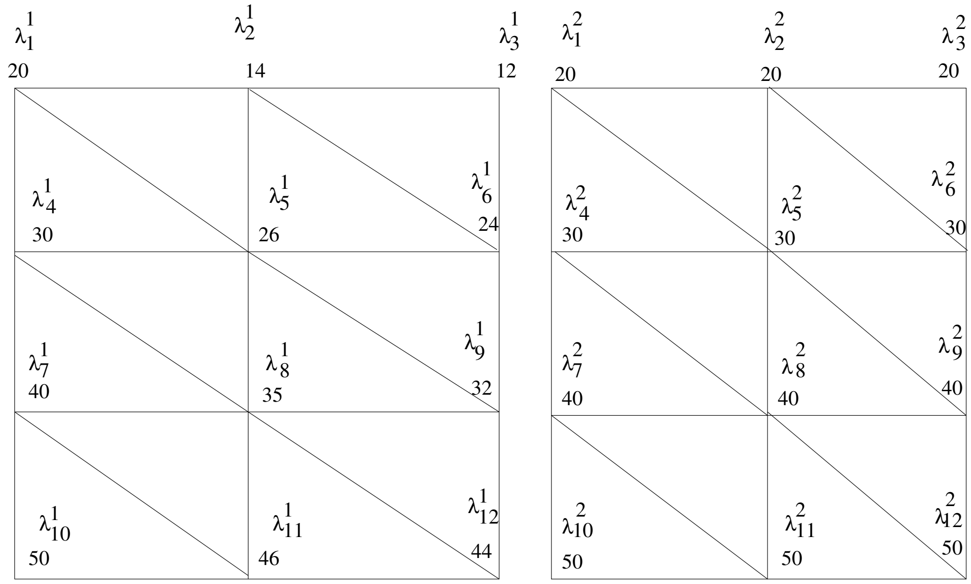

Figure 5.9: Example for comparing vertices and facets

5.3.2 The Construction of Cuts and the Separation Algorithm

The algorithm we have described above can now be used to construct cutting planes for our MIP model. Unfortunately, we do not know the facets that are defining P, since they are relatively difficult to describe even in quite easy situations like the sequence of two pipes described at the beginning of this chapter.

Here we give a example for this situation. Clearly in more complicated or in realistic situations for the Transient Technical Optimisation the problem of describing the facets normally becomes bigger and bigger. As an example let us consider the polyhedron P_Δ in the case which is described in Figure 5.9. According to our algorithm it is very easy to calculate the vertices given in the tables on page 75 and 76 (c.f. also Example 25).

After that (see page 77) we give the complete description of polyhedron P_Δ in this case (the facets have been calculated with the program **Porta Version 1.3**, see [6]).

We remember since the structure of the facets is much more complicated than the structure of the vertices and the complexity usually grows with the complexity of the studied polyhedron it is not very easy to calculate general formulas for some classes of facets which can be useful in a branch-and-cut algorithm.

We remark that in the complete description the first three equations are equal to the three equations in the description of P. The other inequalities (without the inequalities which define non-negativity constraints for $\lambda-$variables) are inequalities resulting from the set/triangle conditions.

λ^1_1	λ^1_2	λ^1_3	λ^1_4	λ^1_5	λ^1_6	λ^1_7	λ^1_8	λ^1_9	λ^1_{10}	λ^1_{11}	λ^1_{12}	λ^2_1	λ^2_2	λ^2_3	λ^2_4	λ^2_5	λ^2_6	λ^2_7	λ^2_8	λ^2_9	λ^2_{10}	λ^2_{11}	λ^2_{12}
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	3/5	0	0	2/5
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	3/ 5	0	0	0	2/ 5
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	3/ 5	0	0	2/ 5	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3/ 5	0	0	0	2/ 5	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3/ 5	0	0	2/ 5	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2/ 5	0	0	3/ 5
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2/ 5	0	0	0	3/ 5
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2/ 5	0	0	3/ 5	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2/ 5	0	0	0	3/ 5	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2/ 5	0	0	3/ 5	0	0
0	0	0	0	0	0	0	0	1/ 3	0	0	2/ 3	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	1/ 3	0	0	2/ 3	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1/ 3	0	0	2/ 3	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	4/ 5	0	0	1/ 5	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	4/ 5	0	0	0	1/ 5	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	4/ 5	0	0	1/ 5	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	4/ 5	0	0	0	1/ 5	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	4/ 5	0	0	1/ 5	0	0	0	0	0
0	0	0	0	0	0	0	4/ 9	0	0	0	5/ 9	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	4/ 9	0	0	0	5/ 9	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	4/ 9	0	0	0	5/ 9	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	6/11	0	0	5/11	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	6/11	0	0	5/11	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	6/11	0	0	5/11	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1/ 2	0	0	1/ 2	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1/ 2	0	0	0	1/ 2	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1/ 2	0	0	1/ 2	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1/ 2	0	0	0	1/ 2	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1/ 2	0	0	1/ 2	0	0	0	0	0
0	0	0	0	0	1/ 4	0	0	3/ 4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1/ 4	0	0	3/ 4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	1/ 4	0	0	3/ 4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	3/ 5	0	0	2/ 5	0	0	0	0	0	0

λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_5^1	λ_6^1	λ_7^1	λ_8^1	λ_9^1	λ_{10}^1	λ_{11}^1	λ_{12}^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2	λ_5^2	λ_6^2	λ_7^2	λ_8^2	λ_9^2	λ_{10}^2	λ_{11}^2	λ_{12}^2
0	0	0	0	0	1	0	0	0	0	0	0	0	3/ 5	0	0	0	2/ 5	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	3/ 5	0	0	2/ 5	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	3/ 5	0	0	0	2/ 5	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	3/ 5	0	0	2/ 5	0	0	0	0	0	0	0	0
0	0	0	0	1/ 3	0	0	0	2/ 3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	1/ 3	0	0	0	2/ 3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1/ 3	0	0	0	2/ 3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	5/ 9	0	0	4/ 9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	5/ 9	0	0	4/ 9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	5/ 9	0	0	4/ 9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	2/ 5	0	0	3/ 5	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	2/ 5	0	0	0	3/ 5	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	2/ 5	0	0	3/ 5	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	2/ 5	0	0	0	3/ 5	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	2/ 5	0	0	3/ 5	0	0	0	0	0	0	0	0
0	0	1/ 3	0	0	2/ 3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	1/ 3	0	0	2/ 3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	1/ 3	0	0	2/ 3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	2/ 5	0	0	0	3/ 5	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	2/ 5	0	0	0	3/ 5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	2/ 5	0	0	0	3/ 5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1/ 2	0	0	1/ 2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1/ 2	0	0	1/ 2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1/ 2	0	0	1/ 2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

EQUALITIES

(1) $-6\lambda_2^1-8\lambda_3^1+10\lambda_4^1+6\lambda_5^1+4\lambda_6^1+20\lambda_7^1+15\lambda_8^1+12\lambda_9^1+30\lambda_{10}^1+26\lambda_{11}^1+24\lambda_{12}^1 -10\lambda_4^2-10\lambda_5^2-10\lambda_6^2-20\lambda_7^2-20\lambda_8^2-20\lambda_9^2-30\lambda_{10}^2-30\lambda_{11}^2-30\lambda_{12}^2 = 0$

(2) $30\lambda_1^1+36\lambda_2^1+38\lambda_3^1+20\lambda_4^1+24\lambda_5^1+26\lambda_6^1+10\lambda_7^1+15\lambda_8^1+18\lambda_9^1+4\lambda_{11}^1+6\lambda_{12}^1-30\lambda_1^2-30\lambda_2^2-30\lambda_3^2-20\lambda_4^2-20\lambda_5^2-20\lambda_6^2-10\lambda_7^2-10\lambda_8^2-10\lambda_9^2 = 0$

(3) $\lambda_1^2+\lambda_2^2+\lambda_3^2+\lambda_4^2+\lambda_5^2+\lambda_6^2+\lambda_7^2+\lambda_8^2+\lambda_9^2+\lambda_{10}^2+\lambda_{11}^2+\lambda_{12}^2 = 1$

INEQUALITIES

(1) $-60\lambda_4^1-270\lambda_7^1-135\lambda_8^1-90\lambda_9^1-580\lambda_{10}^1-432\lambda_{11}^1-378\lambda_{12}^1-90\lambda_2^2-90\lambda_3^2 +60\lambda_6^2+270\lambda_7^2+210\lambda_8^2+210\lambda_9^2+540\lambda_{10}^2+540\lambda_{11}^2+580\lambda_{12}^2 \leq 0$

(2) $-30\lambda_4^1-10\lambda_5^1-110\lambda_7^1-55\lambda_8^1-40\lambda_9^1-265\lambda_{10}^1-191\lambda_{11}^1-154\lambda_{12}^1-20\lambda_2^2-20\lambda_3^2 +30\lambda_6^2+110\lambda_7^2+80\lambda_8^2+80\lambda_9^2+220\lambda_{10}^2+220\lambda_{11}^2+265\lambda_{12}^2 \leq 0$

(3) $-30\lambda_4^1-10\lambda_5^1-110\lambda_7^1-55\lambda_8^1-40\lambda_9^1-240\lambda_{10}^1-176\lambda_{11}^1-154\lambda_{12}^1-20\lambda_2^2-20\lambda_3^2 +30\lambda_6^2+110\lambda_7^2+80\lambda_8^2+80\lambda_9^2+220\lambda_{10}^2+220\lambda_{11}^2+240\lambda_{12}^2 \leq 0$

(4) $-30\lambda_4^1-10\lambda_5^1-110\lambda_7^1-55\lambda_8^1-40\lambda_9^1-220\lambda_{10}^1-176\lambda_{11}^1-154\lambda_{12}^1-20\lambda_2^2-20\lambda_3^2 +30\lambda_6^2+110\lambda_7^2+110\lambda_8^2+80\lambda_9^2+220\lambda_{10}^2+220\lambda_{11}^2+220\lambda_{12}^2 \leq 0$

(5) $-20\lambda_4^1-90\lambda_7^1-45\lambda_8^1-30\lambda_9^1-210\lambda_{10}^1-154\lambda_{11}^1-126\lambda_{12}^1-30\lambda_2^2-30\lambda_3^2 +20\lambda_6^2+90\lambda_7^2+70\lambda_8^2+70\lambda_9^2+180\lambda_{10}^2+180\lambda_{11}^2+210\lambda_{12}^2 \leq 0$

(6) $-20\lambda_4^1-90\lambda_7^1-45\lambda_8^1-30\lambda_9^1-180\lambda_{10}^1-144\lambda_{11}^1-126\lambda_{12}^1-30\lambda_2^2-30\lambda_3^2 +20\lambda_6^2+90\lambda_7^2+90\lambda_8^2+70\lambda_9^2+180\lambda_{10}^2+180\lambda_{11}^2+180\lambda_{12}^2 \leq 0$

(7) $-12\lambda_4^1-90\lambda_7^1-45\lambda_8^1-18\lambda_9^1-212\lambda_{10}^1-144\lambda_{11}^1-126\lambda_{12}^1-18\lambda_2^2-18\lambda_3^2 +12\lambda_6^2+90\lambda_7^2+42\lambda_8^2+42\lambda_9^2+180\lambda_{10}^2+180\lambda_{11}^2+212\lambda_{12}^2 \leq 0$

(8) $-12\lambda_5^1+40\lambda_7^1+15\lambda_8^1+90\lambda_{10}^1+70\lambda_{11}^1+60\lambda_{12}^1 -40\lambda_7^2-40\lambda_8^2-40\lambda_9^2-90\lambda_{10}^2-90\lambda_{11}^2-90\lambda_{12}^2 \leq 0$

(9) $-24\lambda_7^1- 9\lambda_8^1-70\lambda_{10}^1-42\lambda_{11}^1-36\lambda_{12}^1- 6\lambda_4^2+24\lambda_7^2+54\lambda_{10}^2+54\lambda_{11}^2+70\lambda_{12}^2 \leq 0$

(10) $-24\lambda_7^1- 9\lambda_8^1-70\lambda_{10}^1-42\lambda_{11}^1-36\lambda_{12}^1 -6\lambda_5^2-6\lambda_6^2+24\lambda_9^2+70\lambda_{10}^2+70\lambda_{11}^2+54\lambda_{12}^2 \leq 0$

(11) $-15\lambda_7^1-55\lambda_{10}^1-33\lambda_{11}^1-27\lambda_{12}^1-15\lambda_4^2+15\lambda_7^2+45\lambda_{10}^2+45\lambda_{11}^2+55\lambda_{12}^2 \leq 0$

(12) $-15\lambda_7^1-55\lambda_{10}^1-33\lambda_{11}^1-27\lambda_{12}^1 -15\lambda_5^2-15\lambda_6^2+15\lambda_9^2+55\lambda_{10}^2+55\lambda_{11}^2+45\lambda_{12}^2 \leq 0$

(13) $-10\lambda_7^1-45\lambda_{10}^1-27\lambda_{11}^1-18\lambda_{12}^1-10\lambda_4^2+10\lambda_7^2+30\lambda_{10}^2+30\lambda_{11}^2+45\lambda_{12}^2 \leq 0$

(14) $-10\lambda_7^1-45\lambda_{10}^1-27\lambda_{11}^1-18\lambda_{12}^1 -10\lambda_5^2-10\lambda_6^2+10\lambda_9^2+45\lambda_{10}^2+45\lambda_{11}^2+30\lambda_{12}^2 \leq 0$

(15) $10\lambda_7^1+30\lambda_{10}^1+22\lambda_{11}^1+15\lambda_{12}^1 -10\lambda_7^2-10\lambda_8^2-10\lambda_9^2-30\lambda_{10}^2-30\lambda_{11}^2-30\lambda_{12}^2 \leq 0$

(16) $6\lambda_4^1+16\lambda_7^1+11\lambda_8^1+8\lambda_9^1+26\lambda_{10}^1+22\lambda_{11}^1+20\lambda_{12}^1-6\lambda_4^2-6\lambda_5^2-6\lambda_6^2-16\lambda_7^2-16\lambda_8^2-16\lambda_9^2-26\lambda_{10}^2-26\lambda_{11}^2-26\lambda_{12}^2 \leq 0$

(17) $12\lambda_4^1+42\lambda_7^1+27\lambda_8^1+16\lambda_9^1+72\lambda_{10}^1+60\lambda_{11}^1+54\lambda_{12}^1-12\lambda_4^2-12\lambda_5^2-12\lambda_6^2-42\lambda_7^2-42\lambda_8^2-42\lambda_9^2-72\lambda_{10}^2-72\lambda_{11}^2-72\lambda_{12}^2 \leq 0$

(18) $36\lambda_4^1+136\lambda_7^1+81\lambda_8^1+48\lambda_9^1+246\lambda_{10}^1+202\lambda_{11}^1+180\lambda_{12}^1-36\lambda_4^2-36\lambda_5^2-36\lambda_6^2-136\lambda_7^2-136\lambda_8^2-136\lambda_9^2-246\lambda_{10}^2-246\lambda_{11}^2-246\lambda_{12}^2 \leq 0$

(19) $-\lambda_3^1 \leq 0$

(20) $-\lambda_4^1 \leq 0$

(21) $-\lambda_5^1 \leq 0$

(22) $-\lambda_7^1 \leq 0$

(23) $-\lambda_8^1 \leq 0$

(24) $-\lambda_9^1 \leq 0$

(25) $-\lambda_{10}^1 \leq 0$

(26) $-\lambda_{11}^1 \leq 0$

(27) $-\lambda_{12}^1 \leq 0$

(28) $-\lambda_2^2 \leq 0$
(29) $-\lambda_3^2 \leq 0$
(30) $-\lambda_4^2 \leq 0$
(31) $-\lambda_5^2 \leq 0$
(32) $-\lambda_6^2 \leq 0$
(33) $-\lambda_7^2 \leq 0$
(34) $-\lambda_8^2 \leq 0$
(35) $-\lambda_9^2 \leq 0$
(36) $-\lambda_{10}^2 \leq 0$
(37) $-\lambda_{11}^2 \leq 0$
(38) $-\lambda_{12}^2 \leq 0$
(39) $2\lambda_3^1-\lambda_6^1 \leq 0$
(40) $+\lambda_{10}^1 -\lambda_{10}^2-\lambda_{11}^2-\lambda_{12}^2 \leq 0$
(41) $-8\lambda_7^1-3\lambda_8^1-30\lambda_{10}^1-18\lambda_{11}^1-12\lambda_{12}^1-2\lambda_4^2+8\lambda_7^2+18\lambda_{10}^2+18\lambda_{11}^2+30\lambda_{12}^2 \leq 0$
(42) $-8\lambda_7^1-3\lambda_8^1-18\lambda_{10}^1-14\lambda_{11}^1-12\lambda_{12}^1-2\lambda_4^2+8\lambda_7^2+18\lambda_{10}^2+18\lambda_{11}^2+18\lambda_{12}^2 \leq 0$
(43) $-2\lambda_{10}^1-3\lambda_7^2+2\lambda_{10}^2 \leq 0$
(44) $-3\lambda_{10}^1-\lambda_{11}^1-2\lambda_7^2+3\lambda_{10}^2 \leq 0$
(45) $- 8\lambda_7^1- 3\lambda_8^1-30\lambda_{10}^1-18\lambda_{11}^1-12\lambda_{12}^1 -2\lambda_5^2-2\lambda_6^2 +8\lambda_9^2+30\lambda_{10}^2+30\lambda_{11}^2+18\lambda_{12}^2 \leq 0$
(46) $- 8\lambda_7^1- 3\lambda_8^1-18\lambda_{10}^1-14\lambda_{11}^1-12\lambda_{12}^1- 2\lambda_4^2-2\lambda_5^2+8\lambda_7^2+8\lambda_8^2+18\lambda_{10}^2+18\lambda_{11}^2+18\lambda_{12}^2 \leq 0$
(47) $-8\lambda_7^1-3\lambda_8^1-18\lambda_{10}^1-14\lambda_{11}^1-12\lambda_{12}^1 -2\lambda_5^2-2\lambda_6^2 +8\lambda_9^2+18\lambda_{10}^2+18\lambda_{11}^2+18\lambda_{12}^2 \leq 0$
(48) $-3\lambda_{10}^1-\lambda_{11}^1-2\lambda_8^2-2\lambda_9^2+3\lambda_{12}^2 \leq 0$
(49) $-3\lambda_{10}^1-\lambda_{11}^1-2\lambda_7^2-2\lambda_8^2+3\lambda_{10}^2+3\lambda_{11}^2 \leq 0$
(50) $-2\lambda_{10}^1-3\lambda_8^2-3\lambda_9^2+2\lambda_{12}^2 \leq 0$
(51) $-2\lambda_{10}^1-3\lambda_7^2-3\lambda_8^2+2\lambda_{10}^2+2\lambda_{11}^2 \leq 0$
(52) $8\lambda_3^1-10\lambda_4^1-6\lambda_5^1-4\lambda_6^1-20\lambda_7^1-15\lambda_8^1-12\lambda_9^1-30\lambda_{10}^1-26\lambda_{11}^1-24\lambda_{12}^1+10\lambda_4^2+10\lambda_5^2+10\lambda_6^2+20\lambda_7^2+20\lambda_8^2+20\lambda_9^2+30\lambda_{10}^2+30\lambda_{11}^2+30\lambda_{12}^2 \leq 0$
(53) $-4\lambda_5^1+10\lambda_7^1+5\lambda_8^1+20\lambda_{10}^1+16\lambda_{11}^1+14\lambda_{12}^1 -10\lambda_7^2-10\lambda_8^2-10\lambda_9^2-20\lambda_{10}^2-20\lambda_{11}^2-20\lambda_{12}^2 \leq 0$
(54) $-4\lambda_4^1-30\lambda_7^1-15\lambda_8^1-6\lambda_9^1-84\lambda_{10}^1-56\lambda_{11}^1-42\lambda_{12}^1-6\lambda_2^2-6\lambda_3^2+4\lambda_6^2+30\lambda_7^2+14\lambda_8^2+14\lambda_9^2+60\lambda_{10}^2+60\lambda_{11}^2+84\lambda_{12}^2 \leq 0$
(55) $-4\lambda_4^1-30\lambda_7^1-15\lambda_8^1-6\lambda_9^1-60\lambda_{10}^1-48\lambda_{11}^1-42\lambda_{12}^1-6\lambda_2^2-6\lambda_3^2+4\lambda_6^2+30\lambda_7^2+30\lambda_8^2+14\lambda_9^2+60\lambda_{10}^2+60\lambda_{11}^2+60\lambda_{12}^2 \leq 0$
(56) $-4\lambda_4^1-14\lambda_7^1- 9\lambda_8^1-6\lambda_9^1-24\lambda_{10}^1-20\lambda_{11}^1-18\lambda_{12}^1-6\lambda_2^2-6\lambda_3^2+4\lambda_6^2+14\lambda_7^2+14\lambda_8^2+14\lambda_9^2+24\lambda_{10}^2+24\lambda_{11}^2+24\lambda_{12}^2 \leq 0$
(57) $-3\lambda_4^1- \lambda_5^1-20\lambda_7^1-10\lambda_8^1-4\lambda_9^1-58\lambda_{10}^1-38\lambda_{11}^1-28\lambda_{12}^1-2\lambda_2^2-2\lambda_3^2+3\lambda_6^2+20\lambda_7^2+8\lambda_8^2+8\lambda_9^2+40\lambda_{10}^2+40\lambda_{11}^2+58\lambda_{12}^2 \leq 0$
(58) $-3\lambda_4^1- \lambda_5^1-20\lambda_7^1-10\lambda_8^1-4\lambda_9^1-48\lambda_{10}^1-32\lambda_{11}^1-28\lambda_{12}^1-2\lambda_2^2-2\lambda_3^2+3\lambda_6^2+20\lambda_7^2+8\lambda_8^2+8\lambda_9^2+40\lambda_{10}^2+40\lambda_{11}^2+48\lambda_{12}^2 \leq 0$
(59) $-3\lambda_4^1- \lambda_5^1-20\lambda_7^1-10\lambda_8^1-4\lambda_9^1-40\lambda_{10}^1-32\lambda_{11}^1-28\lambda_{12}^1-2\lambda_2^2-2\lambda_3^2+3\lambda_6^2+20\lambda_7^2+20\lambda_8^2+8\lambda_9^2+40\lambda_{10}^2+40\lambda_{11}^2+40\lambda_{12}^2 \leq 0$
(60) $-6\lambda_4^1-2\lambda_5^1-16\lambda_7^1-11\lambda_8^1-8\lambda_9^1-26\lambda_{10}^1-22\lambda_{11}^1-20\lambda_{12}^1-4\lambda_2^2-4\lambda_3^2+6\lambda_6^2+16\lambda_7^2+16\lambda_8^2+16\lambda_9^2+26\lambda_{10}^2+26\lambda_{11}^2+26\lambda_{12}^2 \leq 0$

(61) $4\lambda_7^1+9\lambda_{10}^1+7\lambda_{11}^1+6\lambda_{12}^1-4\lambda_7^2-4\lambda_8^2-4\lambda_9^2-9\lambda_{10}^2-9\lambda_{11}^2-9\lambda_{12}^2 \leq 0$

(62) $4\lambda_3^1+4\lambda_4^1-2\lambda_6^1+14\lambda_7^1+9\lambda_8^1+6\lambda_9^1+24\lambda_{10}^1+20\lambda_{11}^1+18\lambda_{12}^1-4\lambda_4^2-4\lambda_5^2-4\lambda_6^2-14\lambda_7^2-14\lambda_8^2-14\lambda_9^2-24\lambda_{10}^2-24\lambda_{11}^2-24\lambda_{12}^2 \leq 0$

(63) $-5\lambda_8^1+10\lambda_{10}^1+6\lambda_{11}^1-10\lambda_{10}^2-10\lambda_{11}^2-10\lambda_{12}^2 \leq 0$

(64) $-10\lambda_7^1-5\lambda_8^1-2\lambda_9^1-20\lambda_{10}^1-16\lambda_{11}^1-14\lambda_{12}^1+10\lambda_7^2+10\lambda_8^2+10\lambda_9^2+20\lambda_{10}^2+20\lambda_{11}^2+20\lambda_{12}^2 \leq 0$

(65) $-5\lambda_{10}^1-3\lambda_{11}^1-2\lambda_{12}^1+5\lambda_{10}^2+5\lambda_{11}^2+5\lambda_{12}^2 \leq 0$

(66) $-5\lambda_7^1-15\lambda_{10}^1-11\lambda_{11}^1-9\lambda_{12}^1-5\lambda_4^2+5\lambda_7^2+15\lambda_{10}^2+15\lambda_{11}^2+15\lambda_{12}^2 \leq 0$

(67) $-5\lambda_7^1-15\lambda_{10}^1-11\lambda_{11}^1-9\lambda_{12}^1-5\lambda_5^2-5\lambda_6^2+5\lambda_9^2+15\lambda_{10}^2+15\lambda_{11}^2+15\lambda_{12}^2 \leq 0$

(68) $-5\lambda_7^1-15\lambda_{10}^1-11\lambda_{11}^1-9\lambda_{12}^1-5\lambda_4^2-5\lambda_5^2+5\lambda_7^2+5\lambda_8^2+15\lambda_{10}^2+15\lambda_{11}^2+15\lambda_{12}^2 \leq 0$

(69) $5\lambda_7^1-3\lambda_9^1+15\lambda_{10}^1+11\lambda_{11}^1+9\lambda_{12}^1-5\lambda_7^2-5\lambda_8^2-5\lambda_9^2-15\lambda_{10}^2-15\lambda_{11}^2-15\lambda_{12}^2 \leq 0$

(70) $+\lambda_2^2+\lambda_3^2+\lambda_4^2+\lambda_5^2+\lambda_6^2+\lambda_7^2+\lambda_8^2+\lambda_9^2+\lambda_{10}^2+\lambda_{11}^2+\lambda_{12}^2 \leq 1$

(71) $-3\lambda_4^1-\lambda_5^1-20\lambda_7^1-10\lambda_8^1-4\lambda_9^1-58\lambda_{10}^1-38\lambda_{11}^1-28\lambda_{12}^1+2\lambda_2^2+2\lambda_3^2+5\lambda_4^2+2\lambda_5^2+2\lambda_6^2+10\lambda_7^2+10\lambda_8^2+22\lambda_9^2+60\lambda_{10}^2+60\lambda_{11}^2+42\lambda_{12}^2 \leq 2$

(72) $-3\lambda_4^1-\lambda_5^1-20\lambda_7^1-10\lambda_8^1-4\lambda_9^1-48\lambda_{10}^1-32\lambda_{11}^1-28\lambda_{12}^1+2\lambda_2^2+2\lambda_3^2+5\lambda_4^2+2\lambda_5^2+2\lambda_6^2+10\lambda_7^2+10\lambda_8^2+22\lambda_9^2+50\lambda_{10}^2+50\lambda_{11}^2+42\lambda_{12}^2 \leq 2$

(73) $-3\lambda_4^1-\lambda_5^1-20\lambda_7^1-10\lambda_8^1-4\lambda_9^1-40\lambda_{10}^1-32\lambda_{11}^1-28\lambda_{12}^1+2\lambda_2^2+2\lambda_3^2+5\lambda_4^2+2\lambda_5^2+2\lambda_6^2+10\lambda_7^2+10\lambda_8^2+22\lambda_9^2+42\lambda_{10}^2+42\lambda_{11}^2+42\lambda_{12}^2 \leq 2$

(74) $-6\lambda_4^1-2\lambda_5^1-16\lambda_7^1-11\lambda_8^1-8\lambda_9^1-26\lambda_{10}^1-22\lambda_{11}^1-20\lambda_{12}^1+4\lambda_3^2+10\lambda_4^2+10\lambda_5^2+4\lambda_6^2+20\lambda_7^2+20\lambda_8^2+20\lambda_9^2+30\lambda_{10}^2+30\lambda_{11}^2+30\lambda_{12}^2 \leq 4$

(75) $-6\lambda_4^1-2\lambda_5^1-16\lambda_7^1-11\lambda_8^1-8\lambda_9^1-26\lambda_{10}^1-22\lambda_{11}^1-20\lambda_{12}^1+4\lambda_2^2+4\lambda_3^2+10\lambda_4^2+4\lambda_5^2+4\lambda_6^2+20\lambda_7^2+20\lambda_8^2+20\lambda_9^2+30\lambda_{10}^2+30\lambda_{11}^2+30\lambda_{12}^2 \leq 4$

(76) $-2\lambda_3^1+16\lambda_4^1+12\lambda_5^1+10\lambda_6^1+26\lambda_7^1+21\lambda_8^1+18\lambda_9^1+36\lambda_{10}^1+32\lambda_{11}^1+30\lambda_{12}^1-10\lambda_4^2-10\lambda_5^2-10\lambda_6^2-20\lambda_7^2-20\lambda_8^2-20\lambda_9^2-30\lambda_{10}^2-30\lambda_{11}^2-30\lambda_{12}^2 \leq 6$

(77) $-4\lambda_4^1-30\lambda_7^1-15\lambda_8^1-6\lambda_9^1-84\lambda_{10}^1-56\lambda_{11}^1-42\lambda_{12}^1+6\lambda_2^2+6\lambda_3^2+10\lambda_4^2+6\lambda_5^2+6\lambda_6^2+20\lambda_7^2+20\lambda_8^2+36\lambda_9^2+90\lambda_{10}^2+90\lambda_{11}^2+66\lambda_{12}^2 \leq 6$

(78) $-4\lambda_4^1-30\lambda_7^1-15\lambda_8^1-6\lambda_9^1-60\lambda_{10}^1-48\lambda_{11}^1-42\lambda_{12}^1+6\lambda_2^2+6\lambda_3^2+10\lambda_4^2+6\lambda_5^2+6\lambda_6^2+20\lambda_7^2+20\lambda_8^2+36\lambda_9^2+66\lambda_{10}^2+66\lambda_{11}^2+66\lambda_{12}^2 \leq 6$

(79) $-4\lambda_4^1-14\lambda_7^1-9\lambda_8^1-6\lambda_9^1-24\lambda_{10}^1-20\lambda_{11}^1-18\lambda_{12}^1+6\lambda_3^2+10\lambda_4^2+10\lambda_5^2+6\lambda_6^2+20\lambda_7^2+20\lambda_8^2+20\lambda_9^2+30\lambda_{10}^2+30\lambda_{11}^2+30\lambda_{12}^2 \leq 6$

(80) $-4\lambda_4^1-14\lambda_7^1-9\lambda_8^1-6\lambda_9^1-24\lambda_{10}^1-20\lambda_{11}^1-18\lambda_{12}^1+6\lambda_2^2+6\lambda_3^2+10\lambda_4^2+6\lambda_5^2+6\lambda_6^2+20\lambda_7^2+20\lambda_8^2+20\lambda_9^2+30\lambda_{10}^2+30\lambda_{11}^2+30\lambda_{12}^2 \leq 6$

(81) $-12\lambda_4^1-90\lambda_7^1-45\lambda_8^1-18\lambda_9^1-212\lambda_{10}^1-144\lambda_{11}^1-126\lambda_{12}^1+18\lambda_2^2+18\lambda_3^2+30\lambda_4^2+18\lambda_5^2+18\lambda_6^2+60\lambda_7^2+60\lambda_8^2+108\lambda_9^2+230\lambda_{10}^2+230\lambda_{11}^2+198\lambda_{12}^2 \leq 18$

(82) $-30\lambda_4^1-10\lambda_5^1-110\lambda_7^1-55\lambda_8^1-40\lambda_9^1-265\lambda_{10}^1-191\lambda_{11}^1-154\lambda_{12}^1+20\lambda_2^2+20\lambda_3^2+50\lambda_4^2+20\lambda_5^2+20\lambda_6^2+100\lambda_7^2+100\lambda_8^2+130\lambda_9^2+285\lambda_{10}^2+285\lambda_{11}^2+240\lambda_{12}^2 \leq 20$

(83) $-30\lambda_4^1-10\lambda_5^1-110\lambda_7^1-55\lambda_8^1-40\lambda_9^1-240\lambda_{10}^1-176\lambda_{11}^1-154\lambda_{12}^1+20\lambda_2^2+20\lambda_3^2+50\lambda_4^2+20\lambda_5^2+20\lambda_6^2+100\lambda_7^2+100\lambda_8^2+130\lambda_9^2+260\lambda_{10}^2+260\lambda_{11}^2+240\lambda_{12}^2 \leq 20$

(84) $-30\lambda_4^1-10\lambda_5^1-110\lambda_7^1-55\lambda_8^1-40\lambda_9^1-220\lambda_{10}^1-176\lambda_{11}^1-154\lambda_{12}^1+20\lambda_2^2+20\lambda_3^2+50\lambda_4^2+20\lambda_5^2+20\lambda_6^2+100\lambda_7^2+100\lambda_8^2+130\lambda_9^2+240\lambda_{10}^2+240\lambda_{11}^2+240\lambda_{12}^2 \leq 20$

(85) $-20\lambda_4^1-90\lambda_7^1-45\lambda_8^1-30\lambda_9^1-210\lambda_{10}^1-154\lambda_{11}^1-126\lambda_{12}^1+30\lambda_2^2+30\lambda_3^2+50\lambda_4^2+30\lambda_5^2+30\lambda_6^2+100\lambda_7^2+100\lambda_8^2+120\lambda_9^2+240\lambda_{10}^2+240\lambda_{11}^2+210\lambda_{12}^2 \leq 30$

(86) $-20\lambda_4^1-90\lambda_7^1-45\lambda_8^1-30\lambda_9^1-180\lambda_{10}^1-144\lambda_{11}^1-126\lambda_{12}^1+30\lambda_2^2+30\lambda_3^2+50\lambda_4^2+30\lambda_5^2+30\lambda_6^2+100\lambda_7^2+100\lambda_8^2+120\lambda_9^2+210\lambda_{10}^2+210\lambda_{11}^2+210\lambda_{12}^2 \leq 30$

(87) $-60\lambda_4^1-270\lambda_7^1-135\lambda_8^1-90\lambda_9^1-580\lambda_{10}^1-432\lambda_{11}^1-378\lambda_{12}^1+90\lambda_2^2+90\lambda_3^2+150\lambda_4^2+90\lambda_5^2+90\lambda_6^2+300\lambda_7^2+300\lambda_8^2+360\lambda_9^2+670\lambda_{10}^2+670\lambda_{11}^2+630\lambda_{12}^2 \leq 90$

In this example we can already see the increasing complexity of the facets of P_Δ while the complexity of the vertices increases slower. In realistic cases which have to be examined in our MIP-models the facets are much complexer even than in this case.
It is clear from this example that we cannot give an example for a realistic situation in a gas network.
The following table gives an impression how the complexity of the vertices and the facets of P_Δ increases with the increasing number of grid points.
The first column describes the number of triangles (the sum in the in- and outgoing pipe), the second column the number of $\lambda-$variables, the third column describes the number of vertices, the fourth the number facet-defining inequalities and the last column the maximal coefficient within the facet-defining inequalities. The example we have already presented can be found as the case $\Delta = \lambda = 24$.

Δ	λ	vertices	facets	max. coeff.
8	12	16	18	25
12	15	28	26	50
16	18	49	47	42
16	18	41	43	90
24	24	73	90	670
32	32	142	10492	50640

We remark that in our test calculations we need P_Δ with at least $\Delta \geq 80$ and $\lambda \geq 55$. What on earth the facets in such a situation will look like?
We have tested several other examples and usually we got the situation that if we add to P_Δ the first law of Kirchhoff the number of vertices and facets is lower than in the case above but the coefficients of the facets are getting worse (but this cannot be proved in general).

So we cannot yet calculate the facets until now but –blessing in disguise– we have seen that we can calculate the vertices of the polyhedron P. Now it is on time to show what we can do with them.

In order to use the vertices it is first very important to see that in all interesting cases there are only polynomially many of vertices which we can calculate algorithmically in addition.

Lemma 27 *For the polyhedron P (with the usual definitions and notations as used before) exist numbers l, c such that the maximal number of vertices of P is less than or equal to cl^{in+out}.*

Proof. Define a number l^* as

$$l^* := \prod_{j=1}^{in+out} n_j \tag{5.4}$$

where the values $n_j, j = 1, 2, \ldots, in + out$ were defined as the number of subsets in which the set of $\lambda-$variables of the in- and outgoing segments are divided. It is clear that l^* is the number of possible combinations of subsets N_k^i from all in- and outgoing segments were from every segment exactly one subset according to Algorithm 18 is taken. We remark that in the special case P_Δ the values n_j are the number of triangles of the triangulation for the in- and outgoing pipe.
It is necessary for a vertex that the non vanishing $\lambda-$variables belong to exactly one such subset for each segment. Let $m \leq rg(A)$ be the maximal number of non vanishing $\lambda-$variables as it was pointed out in Algorithm 18 (remark that this number already has been calculated). Only in order to blow up the notation not too much we define for $j \in \{1, 2, \ldots, in + out\}$ numbers N_{max}^j:

$$N_{max}^j := max\{|N_1^j|, |N_2^j|, \ldots, |N_{n_j}^j|\}.$$

Then take for $j \in \{1, 2, \ldots, in + out\}$ variables x_j which can be positive (natural) numbers and after that define a number c as:

$$c := \sum_{\sum_{j=1}^{in+out} x_j \leq m} \prod_{j=1}^{in+out} \binom{N_{max}^j}{x_j}$$

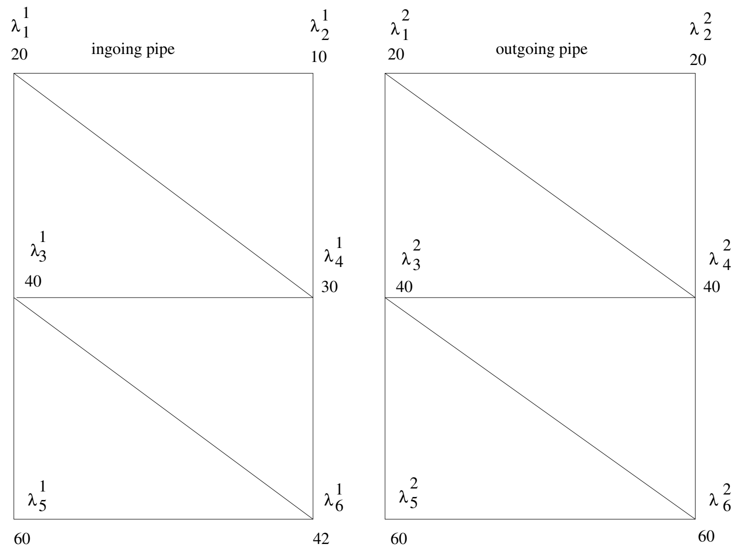

Figure 5.10: Example for comparing vertices and facets

We remark that by construction $\sum_{j=1}^{in+out} x_j \geq in + out$.
The interpretation of c is as follows:
c is an upper bound for the maximal number of possible vertices for the selection of subsets in (4). This is clear because we sum over all selections of $\lambda-$variables (resp. the chosen subsets in $\mathcal{S}$) for which the number of selected $\lambda-$variables is not greater than m. Additionally the product of the binomial coefficients calculates the maximal number of possibilities how we can choose the $\sum_{j=1}^{in+out} x_j$ $\lambda-$variables out of the sets of $\lambda-$variables belonging to the selected feasible subsets. We conclude that the number of vertices cannot be greater than cl^*.
Define

$$l := max\{n_1, n_2, \ldots, n_{in+out}\}.$$

Summarising our argumentation we finally conclude that the number of vertices cannot be greater than cl^{in+out}. $\square$

Note that a trivial upper bound for c is

$$c = 2^{\sum_{j=1}^{in+out} N_{max}^j}.$$

But this value for c is a good deal worse than the (even not quite good) value we have given in the proof of Lemma 27.

Now let in and out be constants. Let also m be a constant which implies (see the last proof) that also c becomes a constant. Then the upper bound in Lemma 27 only depends on l where l describes the number of subdivisions of the grid. We see that in the case of polyhedron P_Δ the polynomiality of Algorithm 18 follows since $m = 3$. Also the polynomiality of Algorithm 18 in the general case of polyhedron P follows. The estimation in the above lemma will be much bigger than the real number of vertices in the polyhedron. For the little example in Figure 5.10 we obtain:

$$c = \binom{3}{1}\binom{3}{1} + \binom{3}{1}\binom{3}{2} + \binom{3}{2}\binom{3}{1} = 27.$$

and $l = 4$, and thus the maximal number of vertices is $27{*}4^{1+1} = 432$. Indeed there are only 16 vertices. Although our estimation is bad it suffices to show that the vertices can be calculated in polynomial time. The number of vertices is usually noticeable lower than the upper bound calculated in Lemma 27. To give a reason for this consider the following

Lemma 28 *Let $\mathcal{S}, \bar{\mathcal{S}}$ be two feasible sets (of $\lambda-$variables) in Algorithm 18 with $\mathcal{S} \subseteq \bar{\mathcal{S}}$. If both sets lead to a vertex of P according to Algorithm 18 they are identical.*

Proof. A vertex of P regarding to $\mathcal{S}$ is the zero-extension of a unique, nonnegative and not vanishing solution of $A_{\mathcal{S}}\lambda_{\mathcal{S}} = b$ (see the description of the algorithm). The same holds for $\bar{\mathcal{S}}$. Adding to the solution of $A_{\mathcal{S}}\lambda_{\mathcal{S}} = b$ the $\lambda-$variables of $\bar{S}^i \setminus S^i$ for all $i \in \{1, 2, \ldots, in + out\}$ which we set to zero. We get a solution of $A_{\bar{\mathcal{S}}}\lambda_{\bar{\mathcal{S}}} = b$. This solution must be the unique solution of $A_{\bar{\mathcal{S}}}\lambda_{\bar{\mathcal{S}}} = b$ by assumption. Analogously we argue when we start from a vertex calculated from $\bar{\mathcal{S}}$. If there would be a vertex belonging to the selection $\mathcal{S}$ we can conclude in the same way as above that the vertices must be equal. $\square$

Lemma 28 has an interesting consequence: If we have found a vertex for a feasible set $\mathcal{S}$ (of a selection of $\lambda-$variables) it is not necessary to search for vertices in a superset of $\mathcal{S}$. Therefore we can start with the feasible sets in which we take exactly one $\lambda-$variable for each segment, i.e., $|S^i| = 1\, \forall i \in \{1, 2, \ldots, in + out\}$, and then look for "bigger" (with respect to set inclusion) feasible sets of selected $\lambda-$variables. In this way we can find all needed vertices in a systematic way.
Another possibility is to start with feasible sets $\mathcal{S}$ for which $|\mathcal{S}| = rg(A)$ holds. If for such a set a vertex is found you do not need to search for a vertex in any subset of this set. This procedure starts from the "biggest" selections whereas the first one starts from the "smallest". In realistic cases (of course you can always construct some pathological cases) this strategy will find the vertices much faster as we have studied in the case of a sequence of two pipes where we modelled the gas flow equation. It turns out that in data sets from gas networks mostly $rg(A) = rg(A_{\mathcal{S}})$ holds for a feasible set $\mathcal{S}$.

For the polyhedron $P_{\triangle}$ Lemma 28 has a nice consequence for the maximal number of vertices:

Lemma 29 *An upper bound for the number of vertices of $P_{\triangle}$ is*

$$9\, n_1\, n_2.$$

Proof. We have described the possibilities for constructing vertices of the polyhedron $P_{\triangle}$. We have seen that for each choice of two triangles there are 9 possibilities for the selection of one $\lambda-$variable from the ingoing pipe and one $\lambda-$variable from the outgoing pipe, i.e., $|S^1| = |S^2| = 1$. Now either this or one of the extensions where we add one $\lambda-$variable either in the chosen triangle of the ingoing pipe or the chosen triangle of the outgoing pipe may result in a vertex, cf Lemma 28. From this argument directly follows Lemma 29. $\square$

Note from Lemma 27 we just obtain a bound of $27\, n_1\, n_2$, since

$$c = \binom{3}{1}\binom{3}{1} + \binom{3}{1}\binom{3}{2} + \binom{3}{2}\binom{3}{1} = 27.$$

Let us come back now to our primal aim.
All the previous ventilations give us the possibility to develop the following **separation algorithm** for P:
Let $v_1, \ldots, v_k$ be the constructed vertices for P (in realistic situations they can be calculated very fast as we have described above).
Let $\bar{\lambda}$ be an optimal LP-solution to be cut off. We look for a cut of the form $a^T\lambda \leq \alpha$ by solving

$$\begin{aligned} z^* = max \quad & a^T\bar{\lambda} - \alpha \\ s.t. \quad & a^T v_i \leq \alpha \quad \text{for } i = 1, \ldots, k. \end{aligned}$$

We remark that w.l.o.g we can assume $\alpha \in \{0, 1, -1\}$.
Let $(\bar{a}, \bar{\alpha})$ be such that $\bar{a}^T\bar{\lambda} - \bar{\alpha} = z^*$.

(a) $\bar{a}^T\lambda \leq \bar{\alpha}$ is valid for P.

Proof. We know from the theory of linear optimisation that every feasible point of the polytope P can be combined as a convex combination of its vertices $v_1, v_2, \ldots, v_k$ (this is correct although P is not convex; we have to remember that for a *nonconvex* polyhedron not every convex combination of a vertex is a point of P but that for every point in P there *exists* a convex combination of vertices of P). That is for $\lambda^* \in P$ there exist nonnegative real numbers $\beta_1, \beta_2, \ldots, \beta_k$ with $\sum_{i=1}^{k} \beta_i = 1$ such that:

$$\lambda^* = \sum_{i=1}^{k} \beta_i v_i.$$

We then calculate:

$$\bar{a}^T \lambda^* = \bar{a}^T \sum_{i=1}^{k} \beta_i v_i = \sum_{i=1}^{k} \beta_i (\bar{a}^T v_i) \leq \sum_{i=1}^{k} \beta_i \bar{\alpha} = \bar{\alpha} \sum_{i=1}^{k} \beta_i = \bar{\alpha}.$$

Therefore $\bar{a}^T \lambda \leq \bar{\alpha}$ is valid for P. □

(b) There exists a violated cut if and only if $z^* > 0$.

Proof. If $z^* > 0$ then due to (a) $\bar{a}^T \lambda \leq \alpha$ is such a cut. On the other hand, suppose $\tilde{a}^T \lambda \leq \tilde{\alpha}$ is a valid inequality violated by $\bar{\lambda}$ then $z^* \geq \tilde{a}^T \bar{\lambda} - \bar{\alpha} > 0$. □

We remark that this algorithm can only separate points outside of conv (P). Since in our case generally $P \subset$ conv (P) (remember that P is not convex) we see that we will not be able to cut off all points that do not fulfil the set conditions. Nevertheless the separation algorithm can be helpful for practical problems.

Additionally we give an easy application of the last proof. Let us assume that the LP-solution fulfils the set condition. Of course in this situation we do not need to use our separation algorithm. This is clear because we know (possibly after a suitable rearrangement of the vertices) that we can write $\bar{\lambda}$ in the form

$$\bar{\lambda} = \sum_{l=n}^{m} \beta_l v_l$$

with $1 \leq n \leq m \leq k$ and $\sum_{l=n}^{m} \beta_l = 1$. In the same way as in the last proof we get $z^* \leq 0$ and so we have shown that in this case expectedly there cannot be a violated cut.

While using the separation algorithm for our test calculations it turned out to be effective to set $\alpha = 1$. So we get a linear program of the form

$$\begin{aligned} \max a^T \bar{\lambda} & \\ A a \ \leq \ \mathbb{1}_{\mathsf{P_V}} & \end{aligned} \tag{LP}$$

where A is a matrix whose number of rows is equal the number of calculated vertices of P, we may call it P_V and the number of columns is $|\mathcal{N}|$.
If we build the dual linear program we get

$$\begin{aligned} \min \mathbb{1}_{\mathsf{P_V}}^{\mathsf{T}} \mathsf{y} & \\ y^T A \ = \ \bar{\lambda}^T & \\ y \ \geq \ 0 & \end{aligned} \tag{DLP}$$

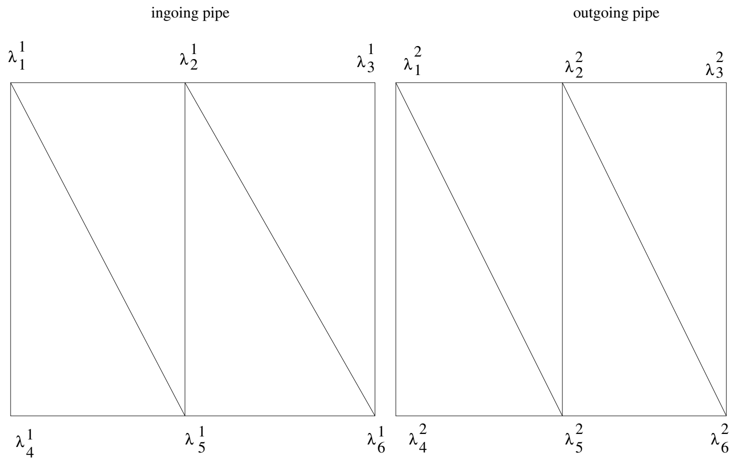

Figure 5.11: Example for separation algorithm

with suitable y.
(DLP) is the `dual linear program` of (LP). From the duality theorem of linear programming we know that (LP) has an optimal solution iff (DLP) has an optimal solution and then these values are equal. This fact can be very important for our problem. If P_V is a good deal bigger than $|\mathcal{N}|$ (DLP) is easier to solve than (LP).
For $\alpha = 0$ or $\alpha = -1$ the situation is analogous.
Let us give an example for the separation algorithm:

Example 30 *As an example let us consider the polyhedron P for one in- and one outgoing pipe with pressure equality in the nodes and first law of Kirchhoff. For the set condition which is described in Figure 5.11 we consider the following polyhedron*

$$\begin{gathered} P = (\lambda_1^1 + \lambda_2^1 + \lambda_3^1 + \lambda_4^1 + \lambda_5^1 + \lambda_6^1 = 1 \\ \lambda_1^2 + \lambda_2^2 + \lambda_3^2 + \lambda_4^2 + \lambda_5^2 + \lambda_6^2 = 1 \\ 20\lambda_1^1 + 14\lambda_2^1 + 12\lambda_3^1 + 30\lambda_4^1 + 26\lambda_5^1 + 24\lambda_6^1 = 20(\lambda_1^1 + \lambda_2^1 + \lambda_3^1) + 30(\lambda_4^1 + \lambda_5^1 + \lambda_6^1) \\ 100\lambda_1^1 + 200\lambda_2^1 + 300\lambda_3^1 + 100\lambda_4^1 + 200\lambda_5^1 + 300\lambda_6^1 = \\ 100\lambda_1^1 + 200\lambda_2^1 + 300\lambda_3^1 + 100\lambda_4^1 + 200\lambda_5^1 + 300\lambda_6^1 \\ \lambda_i^1, \lambda_j^2 \textit{ are nonnegative and satisfy the triangle condition}\}) \end{gathered}$$

Suppose we want to separate the point

$$\bar{\lambda}^T = (0, 0.625, 0, 0.375, 0, 0, 0, 0.6875, 0, 0.3125, 0, 0, 0)$$

which does not fulfil the set condition (see Figure 5.11).
With the formulas we calculated in Remark 26 we get the vertices

Ingoing pipe						*Outgoing pipe*					
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_5^1	λ_6^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2	λ_5^2	λ_6^2
0	0	0	0	1/2	1/2	0	1/2	0	0	0	1/2
0	0	0	2/7	5/7	0	2/7	0	0	0	5/7	0
0	2/5	0	0	0	3/5	0	2/5	3/5	0	0	0
0	0	0	0	0	1	0	0	3/5	0	0	2/5
0	0	0	0	1	0	0	2/5	0	0	3/5	0
0	0	1/3	0	0	2/3	0	0	1	0	0	0
0	1/2	0	0	1/2	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0	0	0	0

We set $\alpha = 1$ in the separation algorithm. So we have to solve the LP:

$$
\begin{array}{ll}
max & 0.625\lambda_2^1 + 0.375\lambda_3^1 + 0.6875\lambda_1^2 + 0.3125\lambda_3^2 \\
s.t. & \\
& \frac{1}{2}\lambda_5^1 + \frac{1}{2}\lambda_6^1 + \frac{1}{2}\lambda_2^2 + \frac{1}{2}\lambda_6^2 \leq 1 \\
& \frac{2}{7}\lambda_4^1 + \frac{5}{7}\lambda_5^1 + \frac{2}{7}\lambda_1^2 + \frac{5}{7}\lambda_5^2 \leq 1 \\
& \frac{2}{5}\lambda_2^1 + \frac{3}{5}\lambda_6^1 + \frac{2}{5}\lambda_2^2 + \frac{3}{5}\lambda_3^2 \leq 1 \\
& \lambda_6^1 + \frac{3}{5}\lambda_3^2 + \frac{2}{5}\lambda_6^2 \leq 1 \\
& \lambda_5^1 + \frac{2}{5}\lambda_2^2 + \frac{3}{5}\lambda_5^2 \leq 1 \\
& \frac{1}{3}\lambda_3^1 + \frac{2}{3}\lambda_6^1 + \lambda_3^2 \leq 1 \\
& \frac{1}{2}\lambda_2^1 + \frac{1}{2}\lambda_5^1 + \lambda_2^2 \leq 1 \\
& \lambda_4^1 + \lambda_4^2 \leq 1 \\
& \lambda_1^1 + \lambda_1^2 \leq 1
\end{array}
$$

We get the cut

$$2\lambda_2^1 + 2\lambda_3^1 + \lambda_4^1 + \lambda_1^2 + \frac{1}{3}\lambda_3^2 \leq 1.$$

It is easy to see that this inequality is valid and cuts off $\bar{\lambda}$.

We remark that another possibility for normalisation of a cut of the form $b^T\lambda \leq \beta$ is to divide this inequality by

$$max(max_{j=1,\ldots,|\mathcal{N}|}|b_j|, |\beta|).$$

So we can look for a cut of the form $a^T\lambda \leq \alpha$ by solving

$$
\begin{array}{rll}
z^* = max & a^T\bar{\lambda} - \alpha & \\
s.t. & a^T v_i \leq \alpha & \text{for } i = 1, \ldots, k \\
& -1 \leq a_j \leq 1 & \text{for } j = 1, \ldots, |\mathcal{N}| \\
& -1 \leq \alpha \leq 1 &
\end{array}
$$

This normalisation can be used if no violated cuts are found for the first discussed normalisation of the right-hand side of the cut. The advantage of this idea is that we do not need to solve a Mixed Integer Program in some cases (since in the first idea usually a cut with $\alpha = 1$ can be found). A little handicap is that the problem has one more variable.

Often the solution time of a problem is too long. In this situation we sometimes can accept that the set conditions are not fulfiled by all pipes or compressors. So we can try to reduce the calculation time by the following idea: In the case that the difference between the linearised pressure at the end of the pipe (that means this value in the LP-solution) and the exact pressure at the calculated point $(p_{in}, q)^T$ of the LP-solution is very small (that means smaller as a value $\epsilon > 0$) we can do without separating the calculated LP-solution. The handling of a compressor is analogous.

5.4 Calculation of the Vertices of P in the Case of Flow Preservation (First Law of Kirchhoff)

As an application of Algorithm 18 we give a little theorem which shows that under certain restrictive conditions the polyhedron which describes the Kirchhoff conservation law in the case of several in - and outgoing pipes has only vertices with binary components. We remember that also a compressor could be taken instead of a pipe. The type of an in- or outgoing segment is of no noteworthy account for the succeeding calculations which is easy to see. Consider again polyhedron P. We only want to model the first law of Kirchhoff: the gas flow preservation. Clearly in this case the matrix A becomes:

$$A = \begin{pmatrix} (\mathbb{1}^1)^T & & & & & & & \\ & (\mathbb{1}^2)^T & & & & & & \\ & & \ddots & & & & & \\ & & & (\mathbb{1}^{\mathsf{in}})^T & & & & \\ & & & & (\mathbb{1}^{\mathsf{in}+1})^T & & & \\ & & & & & (\mathbb{1}^{\mathsf{in}+2})^T & & \\ & & & & & & \ddots & \\ & & & & & & & (\mathbb{1}^{\mathsf{in}+\mathsf{out}})^T \\ (q^1)^T & (q^2)^T & \dots & (q^{in})^T & -(q^{in+1})^T & -(q^{in+2})^T & \dots & -(q^{in+out})^T \end{pmatrix}$$

since we easily can forget the pressure conditions in this case. in and out again describe the number of in - and outgoing pipes. We assume $in \geq 1$ and $out \geq 1$. The vector b becomes

$$b = \begin{pmatrix} \mathbb{1}_{\mathsf{in}+\mathsf{out}} \\ 0 \end{pmatrix} \in \mathbb{R}^{in+out+1}.$$

We will assume now that for all pipes the gas flow is discretisised by the same equidistant discretisation. That means we define for all pipes (in - and outgoing) real numbers a and b with $b > a \geq 0$ for the minimal gas flow a, the maximal gas flow b through the pipe and a natural number n such that holds:

$$q^1 = q^2 = \cdots = q^{in} = q^{in+1} = \cdots = q^{in+out} = \begin{pmatrix} q \\ \vdots \\ q \end{pmatrix}$$

with a vector q of the form

$$q = \begin{pmatrix} a \\ a + \frac{b-a}{n} \\ a + 2\frac{b-a}{n} \\ \vdots \\ a + i\frac{b-a}{n} \\ \vdots \\ a + (n-1)\frac{b-a}{n} \\ b \end{pmatrix}$$

We assume the triangulation to be in the form which is shown in Figure 5.2 or 5.3.

Theorem 31 *If*

$$n\ \frac{a\,|out - in|}{b - a} \in \mathbb{Z}$$

than all vertices of P (with matrix A and vector b as defined above) are elements of $\{0,1\}^{\mathcal{N}}$.

Proof. For the proof we use our algorithm for the calculation of the vertices of P (and of course the notation we used there).

We can easy calculate $rg(A) = in + out + 1$. So we only need to consider feasible sets $\mathcal{S}$ with $in + out \leq |\mathcal{S}| \leq in + out + 1$. Consequently we examine the following two cases:

First case: $|\mathcal{S}| = in + out$.
We know that if we get a solution of $A_{\mathcal{S}}\lambda_{\mathcal{S}} = b$ for every pipe at least one λ-variable must be greater than zero. We select $in + out$ variables (for each pipe one λ-variable) and from the definition of A it is clear that all these variables must be 1. So if we get a vertex in this case, it is clear that it is element of $\{0,1\}^{\mathcal{N}}$.

Second case: $|\mathcal{S}| = in + out + 1$.
It is well known that in this case there exists exactly one pipe for which we select two λ-variables. We again examine two cases:
In the first case of the situation $|\mathcal{S}| = in + out + 1$ let w.l.o.g. be the first ingoing pipe that for which we select two λ-variables. Here $A_{\mathcal{S}}\lambda_{\mathcal{S}} = b$ reduces to:

$$\lambda^1_{i_1} q^1_{i_1} + \lambda^1_{i_1+1} q^1_{i_1+1} + \sum_{k=2}^{in} \lambda^k_{i_k} q^k_{i_k} = \sum_{k=in+1}^{in+out} \lambda^k_{i_k} q^k_{i_k}$$

with

$$\lambda^1_{i_1} + \lambda^1_{i_1+1} = 1$$

and

$$\lambda^k_{i_k} = 1 \qquad \forall k \in \{2, \ldots, in + out\}$$

where $\lambda^k_{i_k}$ means that we have selected a λ-variable i_k for pipe k for that the gas flow value $q^k_{i_k}$ is equal to $a + i_k \frac{b-a}{n}$. Note that in the term $\lambda^1_{i_1} q^1_{i_1} + \lambda^1_{i_1+1} q^1_{i_1+1}$ as a consequence of the structure of the discretisation $q^1_{i_1} = a + i_1 \frac{b-a}{n}$ and $q^1_{i_1+1} = a + (i_1 + 1)\frac{b-a}{n}$ must hold since in the case $q^1_{i_1} = q^1_{i_1+1}$ the linear equality system $A_{\mathcal{S}}\lambda_{\mathcal{S}} = b$ does not have a unique solution which is easy to see. We can rewrite the above conditions as

$$\lambda^1_{i_1} q^1_{i_1} + (1 - \lambda^1_{i_1}) q^1_{i_1+1} + \sum_{k=2}^{in} q^k_{i_k} = \sum_{k=in+1}^{in+out} q^k_{i_k}.$$

We get

$$(q^1_{i_1} - q^1_{i_1+1})\lambda^1_{i_1} = \sum_{k=in+1}^{in+out} q^k_{i_k} - \sum_{k=2}^{in} q^k_{i_k} - q^1_{i_1+1}.$$

With $q^k_{i_k} = a + i_k \frac{b-a}{n}$ we conclude

$$(a + i_1 \frac{b-a}{n} - (a + (i_1 + 1)\frac{b-a}{n}))\lambda^1_{i_1} =$$

$$\sum_{k=in+1}^{in+out} (a + i_k \frac{b-a}{n}) - \sum_{k=2}^{in} (a + i_k \frac{b-a}{n}) - (a + (i_1 + 1)\frac{b-a}{n}).$$

With a short calculation this equation can be reduced to

$$-\frac{b-a}{n}\lambda^1_{i_1} = (\sum_{k=in+1}^{in+out} 1 - \sum_{k=1}^{in} 1)a + \frac{b-a}{n}(\sum_{k=in+1}^{in+out} i_k - \sum_{k=1}^{in} i_k) - \frac{b-a}{n}$$

which is apparently equivalent to

$$\lambda_{i_1}^1 = -n\frac{a(out - in)}{b-a} + \sum_{k=1}^{in} i_k - \sum_{k=in+1}^{in+out} i_k + 1.$$

Now w.l.o.g. we consider the case of the situation $|\mathcal{S}| = in + out + 1$ that we have selected two λ-variables for the first outgoing pipe. The calculation is quite the same as we have done for the first ingoing pipe. Here $A_{\mathcal{S}}\lambda_{\mathcal{S}} = b$ reduces to:

$$\sum_{k=1}^{in} \lambda_{i_k}^k q_{i_k}^k = \lambda_{i_{in+1}}^{in+1} q_{i_{in+1}}^{in+1} + \lambda_{i_{in+1}+1}^{in+1} q_{i_{in+1}+1}^{in+1} + \sum_{k=in+2}^{in+out} \lambda_{i_k}^k q_{i_k}^k$$

with

$$\lambda_{i_{in+1}}^{in+1} + \lambda_{i_{in+1}+1}^{in+1} = 1$$

and

$$\lambda_{i_k}^k = 1 \qquad \forall k \in \{1, \ldots, in, in+2, \ldots, in+out\}$$

where again $\lambda_{i_k}^k$ means that we have selected a λ-variable i_k for pipe k for that the gas flow value $q_{i_k}^k$ is equal to $a + i_k\frac{b-a}{n}$. Of course we get $q_{i_{in+1}}^{in+1} = a + i_{in+1}\frac{b-a}{n}$ and $q_{i_{in+1}+1}^{in+1} = a + (i_{in+1}+1)\frac{b-a}{n}$. We conclude as new condition

$$\sum_{k=1}^{in} q_{i_k}^k = \lambda_{i_{in+1}}^{in+1} q_{i_{in+1}}^{in+1} + (1 - \lambda_{i_{in+1}}^{in+1}) q_{i_{in+1}+1}^{in+1} + \sum_{k=in+2}^{in+out} q_{i_k}^k.$$

With $q_{i_k}^k = a + i_k\frac{b-a}{n}$ we calculate

$$(a + i_{in+1}\frac{b-a}{n} - (a + (i_{in+1}+1)\frac{b-a}{n}))\lambda_{i_{in+1}}^{in+1} =$$

$$\sum_{k=1}^{in}(a + i_k\frac{b-a}{n}) - \sum_{k=in+2}^{in+out}(a + i_k\frac{b-a}{n}) - (a + (i_{in+1}+1)\frac{b-a}{n}).$$

We get after a longer calculation

$$\lambda_{i_{in+1}}^{in+1} = -n\frac{a(in - out)}{b-a} + \sum_{k=in+1}^{in+out} i_k - \sum_{k=1}^{in} i_k + 1.$$

Summarising our last ventilations we see that if

$$n\ \frac{a\,|out - in|}{b-a} \in \mathbb{Z}$$

we get because of $0 \le \lambda_{i_k}^k \le 1$ for all $k \in \{1, 2, \ldots, in + out\}$ that $\lambda_{i_k}^k \in \{0, 1\}$ for every potential vertex of P. In fact we do not get any new vertex in the case $|\mathcal{S}| = in + out + 1$ and thus our proof is complete. □

From Theorem 31 we get the following

Corollary 32 *Assume an equidistant discretisation for all segements. Under the requirements of Theorem 31 holds:*

- *If $in = out$ than all vertices of P are elements of $\{0,1\}^{\mathcal{N}}$.*

- *If* $a = 0$ *than all vertices of* P *are elements of* $\{0,1\}^{\mathcal{N}}$.
- *If* $b = 2a$ *than all vertices of* P *are elements of* $\{0,1\}^{\mathcal{N}}$.

Proof. Obvious. □

Here is a further remark on Theorem 31:

Remark 33 *We give a simple example in the case* $in \neq out$, $a \neq 0$ **and** $b \neq 2a$. *Take* $a = 10, b = 50, n = 2, in = 1, out = 2$. *Here we see* $\frac{2 \cdot 10 \cdot |2-1|}{50-10} = \frac{1}{2}$. *Because of* $\frac{1}{2} \cdot 10 + \frac{1}{2} \cdot 30 + 1 \cdot 30 = 50$ *we see that we indeed get a vertex whose components are not in* $\{0,1\}$.

Lemma 34 *We consider polyhedron* P *as defined in this section with*

$$n \, \frac{a\,|out - in|}{b-a} \in \mathbb{Z}.$$

We define a polyhedron P_S *as follows:*
The matrix A *and the vector* b *have the same form as for polyhedron* P *in this situation. But we define* P_S *so that every gas flow value only exists once for each in- and outgoing segment. We are going to precise this definition. Formally that means that*

$$q^1 = q^2 = \cdots = q^{in} = q^{in+1} = \cdots = q^{in+out} = q$$

holds with q being the well known vector of the form

$$q = \begin{pmatrix} a \\ a + \frac{b-a}{n} \\ a + 2\frac{b-a}{n} \\ \vdots \\ a + i\frac{b-a}{n} \\ \vdots \\ a + (n-1)\frac{b-a}{n} \\ b \end{pmatrix}$$

Then the following facts hold:

- *The vertices of* P *can be calculated directly from the vertices of* P_S. *Therefore we only need to take a vertex of* P_S *and get a new vertex by setting the other* $\lambda-$*variables of* P *to zero. After that for each in- and outgoing segment alternate set the* $\lambda-$*variables with the same gas flow value to one and do this procedure over all possibilities for all segments.*
- *Let* $\hat{a}^T\lambda \leq \alpha$ *be a valid inequality of* P_S. *Then* $\bar{a}^T\bar{\lambda} \leq \alpha$ *is a valid inequality of* P *if we define* $\bar{a}$ *in the following way such that the vector* $\bar{a}$ *is a direct extension of vector* $\hat{a}$*:*
  For every segment and for every $\lambda-$*variable belonging to a special gas flow value the value of* $\bar{a}$ *is equal to the value of* $\hat{a}$, *that means* $\bar{a}$ *can be written in the form*

$$\bar{a} = \begin{pmatrix} \hat{a} \\ \hat{a} \\ \vdots \\ \hat{a} \end{pmatrix}$$

  where the number of vectors $\hat{a}$ *depends on the set condition.*

- *If $\hat{a}^T\lambda \leq \alpha$ is a facet than $\bar{a}^T\bar{\lambda} \leq \alpha$ is a facet, too. This means if we know the complete description of P_S we can calculate the complete description of P.*

The consequence of this lemma is that for a practical calculation and the use of the separation algorithm we only have to calculate facets or valid inequalities for P_S. But we add that this situation is so theoretical that it will not be helpful for calculations in real-world cases.

Proof. The proofs of the facts are very simple and so we omit them. □

We give a short remark for the calculation of vertices that gives additional information for Remark 26 in the case of polyhedron $P_\triangle$ with flow pressure conservation.

Remark 35 *Let us consider polyhedron $P_\triangle$ added by the first law of Kirchhoff. We assume that the in- and the outgoing pipe have the same equidistant discretisation which we have defined in order to formulate Theorem 31. Then the points calculated in 5 and 6 on page 73 do not lead to a new vertex. In case 4 we may have selected $\lambda_i^1, \lambda_j^1, \lambda_k^2, \lambda_l^2$ as described on page 73. Then we can only get a vertex if $p_l^2 - p_j^1 \geq 0$ and $p_i^1 - p_k^2 \geq 0$ with:*

$$\lambda_i^1 = \lambda_k^2 = \frac{p_l^2 - p_j^1}{p_i^1 - p_j^1 - p_k^2 + p_l^2},$$

$$\lambda_j^1 = \lambda_l^2 = \frac{p_i^1 - p_k^2}{p_i^1 - p_j^1 - p_k^2 + p_l^2}.$$

Thus we can restrict us to 1,2,3 on page 73 in order to calculate the vertices of this polyhedron. The proofs of these facts are quite simple and analogously to proof 5.4 and so we omit them.

Chapter 6

Cutting Planes via Lifting

6.1 Facets or Valid Inequalities for small Triangulations and Lifting

In order to find a better description of the studied polyhedron we try to lift facets or valid inequalities of small subproblems to complexer situations. Since the pressure equality at each node for in - and outgoing pipes which can be described by the polyhedron P_Δ is found very often in every model we have calculated the vertices and facets or valid inequalities in the most important cases for this situation for small discretisations. These facets or valid inequalities are to be found in the Appendix. In the next subsections we show in which way we try to find valid inequalities in the case of complexer discretisations by lifting the calculated facets or valid inequalities.

The following ventilations, formulas and algorithms are applicable to the more general polyhedron P. The valid inequalities and facets for small triangulations have been calculated for the special polyhedron P_Δ (see Appendix). So all theoretical results for polyhedron P in this chapter can be applied in the case of polyhedron P_Δ.

The sense of this chapter is as follows:
Principally we have already developed a suitable branch-and-cut algorithm for the general case of polyhedron P in the last chapter (for the description of branching see chapter 7) and if we are lucky in this way we have found a powerful method in order to approximate the nonlinearities of our model. While solving the LP-relaxation of our problem we are often able to calculate an inequality that cuts of the solution of the LP-Relaxation. But we do not know anything about the dimension of this inequality (mostly this cut will not induce a facet). The calculation of facets or valid inequalities for P is very complex (we have seen an example in Chapter 5). Since the knowledge of facets or valid inequalities of P will be very helpful in order to tighten the MIP-formulation of our model (and so to fasten the calculation) we start with small cases (small triangulations) and lift in the new variables of complexer situations. The basic foundations of lifting techniques can be found in [32], [48]. Lifting has been helpful in several practical problems like in the travelling salesman problems or in the optimisation of Steiner trees (see e.g. [27], [28]).
In our special situation we are dealing with continuous lifting.

6.2 The general Lifting Algorithm for Polyhedron P

Let us first discuss an idea for sequential lifting in the case of polyhedron P for which we know the vertices.
Define a set S of indices of $\lambda-$variables of $P \subseteq \mathbb{R}^N$ with $\mathbb{N} \ni |N| < \infty$ and $S \subset N$. We define a polyhedron P' by

$$P' = P \cap \{\lambda_i = 0 \mid \forall i \in S\}$$

which roughly speaking means that P' is the polyhedron belonging to a "smaller" set condition that is contained in P defined by its set condition (see page 61). Formally it also holds $P' \subseteq \mathbb{R}^N$.

For the following ventilations at first let $|S| = 1$. Let $a^T\lambda \leq \alpha$ with $a, \lambda \in \mathbb{R}^N$ with $a_S = 0$ be a facet (or a valid inequality) of P'. We want to lift this inequality to a valid inequality of P. In words that means we look at the "complexer triangulation" (belonging to P) which "contains" the "smaller triangulation" (belonging to P'). We call $vert(P) = \{\lambda^1, \lambda^2, \ldots\}$ with $vert(P) < \infty$ the set of all vertices of P (where all vertices are elements of $\mathbb{R}^N$) and analogously $vert(P') = \{(\lambda')^1, (\lambda')^2, \ldots\}$ the set of vertices of P'. We want to "lift the variable λ_S" (in $a^T\lambda \leq \alpha$) which formally means:

Calculate a vector $b \in \mathbb{R}^N$ with $b_{N\setminus S} = 0_{|N|-|S|}(= 0_{|N|-1})$ such that

$$b^T\lambda + a^T\lambda \leq \alpha$$

is valid for P.

We remark that $\lambda_S^v \in \mathbb{R}$ is the value of the vertex $\lambda^v \in vert(P)$ for which we want to lift in the only non-vanishing component in b (which is b_S).

Theorem 36 *Define*

$$b_S = min\{\frac{\alpha - a^T\lambda^v}{\lambda_S^v} | \forall \lambda^v \in vert(P) \text{ with } \lambda_S^v \neq 0\}, \text{ if } \ |\{\lambda^v \in vert(P) \mid \lambda_S^v \neq 0\}| \neq 0$$

and set

$$b_S = 0, \text{ if } \ |\{\lambda^v \in vert(P) \mid \lambda_S^v \neq 0\}| = 0.$$

Then $b_S \geq 0$. The inequality

$$b^T\lambda + a^T\lambda \leq \alpha$$

is valid for P.

Proof. Since we know the vertices of P this proof is very simple:
We want to calculate b_S such that holds:

$$b^T\lambda^v + a^T\lambda^v = b_S\lambda_S^v + a^T\lambda^v \leq \alpha \qquad \forall \lambda^v \in vert(P).$$

Then we get a valid inequality for P. (The proof for this fact is exactly the same as the proof on page 83).

We first examine the case $|\{\lambda^v \in vert(P) | \lambda_S^v \neq 0\}| = 0$. Take a $\lambda^{\bar{v}} \in vert(P)$. Then $b_S = 0$ and we get

$$b^T\lambda^{\bar{v}} + a^T\lambda^{\bar{v}} = b_S\lambda_S^{\bar{v}} + a^T\lambda^{\bar{v}} = a^T\lambda^{\bar{v}} \leq \alpha.$$

The last inequality holds because it is easy to calculate that for all $\lambda^{\bar{v}} \in vert(P)$ also holds $\lambda^{\bar{v}} \in vert(P')$.

Now let $|\{\lambda^v \in vert(P) | \lambda_S^v \neq 0\}| \neq 0$. We first select a $\lambda^{\bar{v}} \in vert(P)$ with $\lambda_S^{\bar{v}} = 0$. It is easy to see that the argumentation is completely analogous as in the situation we discussed above.

Finally let $|\{\lambda^v \in vert(P)|\lambda_S^v \neq 0\}| \neq 0$ and select a $\lambda^{\bar{v}} \in vert(P)$ with $\lambda_S^{\bar{v}} \neq 0$. We calculate

$$\begin{aligned}
& b^T\lambda^{\bar{v}} + a^T\lambda^{\bar{v}} \\
&= b_S\lambda_S^{\bar{v}} + a^T\lambda^{\bar{v}} \\
&= min\{\tfrac{\alpha - a^T\lambda^v}{\lambda_S^v} | \forall\lambda^v \in vert(P) \text{ with } \lambda_S^v \neq 0\}\lambda_S^{\bar{v}} + a^T\lambda^{\bar{v}} \\
&\leq \tfrac{\alpha - a^T\lambda^{\bar{v}}}{\lambda_S^{\bar{v}}}\lambda_S^{\bar{v}} + a^T\lambda^{\bar{v}} \\
&= \alpha - a^T\lambda^{\bar{v}} + a^T\lambda^{\bar{v}} \\
&= \alpha.
\end{aligned}$$

Now the proof is complete.
We remark that b_S is really a minimum and not a infimum since $|vert(P)| < \infty$. □

From Theorem 36 we get the following

Corollary 37 *Is $a^T\lambda \leq \alpha$ a facet of P' then $b^T\lambda + a^T\lambda \leq \alpha$ is a facet of P.*

Proof.
Consider again the case $|\{\lambda^v \in vert(P)|\lambda_S^v \neq 0\}| = 0$. It is easy to see that in this case holds $vert(P') = vert(P)$. So the dimension of P' is equal to the dimension of P and so the lifted inequality is also a facet.

Now let $|\{\lambda^v \in vert(P)|\lambda_S^v \neq 0\}| \neq 0$. In this case we know that there exists a vertex $\lambda^{\bar{v}} \in vert(P)$ such that

$$b_S = \frac{\alpha - a^T\lambda^{\bar{v}}}{\lambda_S^{\bar{v}}}.$$

For this $\lambda^{\bar{v}}$ we calculate

$$a^T\lambda^{\bar{v}} + \frac{\alpha - a^T\lambda^{\bar{v}}}{\lambda_S^{\bar{v}}}\lambda_S^{\bar{v}} = a^T\lambda^{\bar{v}} + \alpha - a^T\lambda^{\bar{v}} = \alpha.$$

So it is clear that the number of vertices that fulfil the inequality at equality increases at least by one. In order to show that the facet of P' is lifted to a facet of P we notice that the new vertices of P are affinely independent from the (zero-extension of the) vertices of P'. This is sufficient in order to show that a facet of P' is lifted to a facet of P (the dimension of P' and P is irrelevant for this fact). □

Let us marginally describe in the general case of polyhedron P a second possibility for a lifting algorithm in the situation that we do **not** know the vertices of P :
Again let $a^T\lambda \leq \alpha$ (belonging to P') be a facet or a valid inequality of a smaller triangulation P' of P. The further assumptions are quite the same as we have used in Theorem 36. We again want to lift in variable λ_S (which means to calculate b_S). Assume we are able to solve the following LP

$$\begin{aligned} z^* = max \quad & a^T\lambda \\ s.t. \quad & \lambda \in P. \end{aligned}$$

Theorem 38

(a) If $a^T\lambda \leq \alpha$ is valid for P we get another valid inequality for P of the form

$$a^T\lambda + |z^* - \alpha|\lambda_S \leq max\{z^*, \alpha\}.$$

(b) If $a^T\lambda \le \alpha$ is not valid for P we get a valid inequality for P of the form

$$a^T\lambda + |z^* - \alpha|\lambda_S \le max\{2z^* - \alpha, \alpha\}.$$

The inequality in (b) is weaker than the inequality in (a).

Proof. We know $\lambda_S \le 1$.

(a) Is $z^* \ge \alpha$ we calculate

$$a^T\lambda + |z^* - \alpha|\lambda_S \le \alpha + (z^* - \alpha) = z^* = max\{z^*, \alpha\}.$$

Here we used that $a^T\lambda \le \alpha$ is valid for P.
Is $z^* < \alpha$ we calculate

$$a^T\lambda + |z^* - \alpha|\lambda_S \le z^* - (z^* - \alpha) = \alpha = max\{z^*, \alpha\}.$$

(b) Is $z^* \ge \alpha$ we calculate

$$a^T\lambda + |z^* - \alpha|\lambda_S \le z^* + (z^* - \alpha) = 2z^* - \alpha = max\{2z^* - \alpha, \alpha\}.$$

Remember that $a^T\lambda \le \alpha$ is not valid in this case.
Is $z^* < \alpha$ we calculate

$$a^T\lambda + |z^* - \alpha|\lambda_S \le z^* - (z^* - \alpha) = \alpha = max\{2z^* - \alpha, \alpha\}.$$

Additionally we calculate

$$max\{2z^* - \alpha, \alpha\} \ge max\{z^*, \alpha\}$$

since from $z^* \ge \alpha$ follows $2z^* - \alpha \ge z^*$ and for $z^* < \alpha$ both maxima are equal to α and so the inequality in (b) is weaker than the inequality in (a). $\square$

For the calculation of z^* we only need to test the vertices of P if these are known. It is also possible to solve the LP for each polyhedron whose union is P and then take the minimum of all solutions. Since Theorem 38 also holds if the vertices of P are not known we have formulated Theorem 38 as done above.

6.3 Example for Lifting

In the situation described by Figure A.5 we select Case 15 (see page 144 in Appendix): We want to show that we can get the facet

$$\frac{\alpha\delta - \alpha\beta}{\beta(\alpha\delta - \alpha - \delta)}\lambda_3^1 - \frac{\alpha}{\alpha\delta - \alpha - \delta}\lambda_4^1 + \frac{\delta}{\alpha + \delta - \alpha\delta}(\lambda_3^2 + \lambda_4^2) \le 1.$$

in Case 15 by the lifting algorithm described in Theorem 36: Let us begin with the second facet in Case 9 of polyhedron $P_\triangle$ according to Figure A.4 (see page 132 in Appendix) as starting inequality for the lifting process. This inequality reads for polyhedron $P_\triangle$ in Case 15 as

$$\frac{\alpha}{\alpha + \delta - \alpha\delta}\lambda_4^1 + \frac{\delta}{\alpha + \delta - \alpha\delta}\lambda_4^2 \le 1.$$

First we want to lift variable λ_3^2. According to the lifting algorithm let μ be the coefficient of variable λ_3^2. We want to construct a valid inequality for $P_\triangle$ of the form

$$\frac{\alpha}{\alpha + \delta - \alpha\delta}\lambda_4^1 + \mu\lambda_3^2 + \frac{\delta}{\alpha + \delta - \alpha\delta}\lambda_4^2 \le 1.$$

Remember that we have to check the following 13 vertices of the polyhedron $P_\triangle$ in Case 15:

Ingoing pipe				Outgoing pipe				
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2	
0	0	0	1	0	α	0	$1-\alpha$	1
0	0	0	1	α	0	0	$1-\alpha$	2
0	0	0	1	α	0	$1-\alpha$	0	3
0	0	β	$1-\beta$	0	0	0	1	4
0	0	β	$1-\beta$	0	0	1	0	5
0	γ	0	$1-\gamma$	0	1	0	0	6
0	γ	0	$1-\gamma$	1	0	0	0	7
δ	0	0	$1-\delta$	0	0	0	1	8
δ	0	0	$1-\delta$	0	0	1	0	9
ϵ	$1-\epsilon$	0	0	0	1	0	0	10
ϵ	$1-\epsilon$	0	0	1	0	0	0	11
ζ	$1-\zeta$	0	0	0	0	0	1	12
ζ	$1-\zeta$	0	0	0	0	1	0	13

Fortunately the calculation for the vertices $1, 2, 4, 6, 7, 8, 10, 11, 12$ does not lead to any complication (calculate the value of the left-hand side of the inequality for these vertices). From the calculation for the vertices $3, 5, 9, 13$ we get from the algorithm that we have to calculate

$$min\{1, \frac{\delta}{\alpha+\delta-\alpha\delta}, 1+\frac{\alpha}{\alpha\delta-\alpha-\delta}(1-\beta), \frac{1}{1-\alpha}(1+\frac{\alpha}{\alpha\delta-\alpha-\delta})\}.$$

We calculate (compare proof of Lemma 45)

$$\frac{\delta}{\alpha+\delta-\alpha\delta} < \frac{\delta}{\delta\alpha+\delta-\alpha\delta} = 1.$$

Furthermore it is clear that

$$\frac{1}{1-\alpha}(1+\frac{\alpha}{\alpha\delta-\alpha-\delta}) = \frac{\delta}{\alpha+\delta-\alpha\delta}.$$

Since for the polyhedron in Case 15 also $\beta > \delta$ must hold we conclude

$$1-\beta < 1-\delta \Rightarrow \delta+\alpha(1-\beta) < \alpha+\delta-\alpha\delta \Rightarrow \frac{\delta}{\alpha+\delta-\alpha\delta} < 1-\frac{\alpha(1-\beta)}{\alpha+\delta-\alpha\delta}.$$

Summarising these calculations we get

$$\mu = \frac{\delta}{\alpha+\delta-\alpha\delta}$$

and we have calculated a valid inequality of the form

$$\frac{\alpha}{\alpha+\delta-\alpha\delta}\lambda_4^1 + \frac{\delta}{\alpha+\delta-\alpha\delta}(\lambda_3^2+\lambda_4^2) \leq 1.$$

Now we are able to lift in variable λ_3^1: Since we have a valid inequality now we notice that we only have to consider vertices 4 and 5. We see at one glance that the lifting coefficient for λ_3^1 is $\frac{\alpha\delta-\alpha\beta}{\beta(\alpha\delta-\alpha-\delta)}$. The new valid inequality therefore is

$$\frac{\alpha\delta-\alpha\beta}{\beta(\alpha\delta-\alpha-\delta)}\lambda_3^1 - \frac{\alpha}{\alpha\delta-\alpha-\delta}\lambda_4^1 + \frac{\delta}{\alpha+\delta-\alpha\delta}(\lambda_3^2+\lambda_4^2) \leq 1$$

which is exactly the facet we wanted to construct by lifting.

6.4 A Separation Algorithm via Lifting small Facets or Valid Inequalities

In our recent preliminaries we showed how to lift the facets or valid inequalities of smaller triangulations in order to get valid inequalities of complexer triangulations. We can add these inequalities at the beginning of the calculations in order to tighten the formulation of our model. The better way is to construct the lifted inequalities such that they can be used for a separation algorithm. We get the following **separation algorithm** for P which is based on simultaneous lifting using our knowledge of the vertices. In this case simultaneous lifting becomes very easy (see [48]):

Let $vert(P)$ again be the set of vertices of P which we have already calculated.
Let $\bar{\lambda}$ be an optimal LP-solution of the relaxation of P to be cut off. In this subsection we omit the assumption of the beginning of this chapter that $|S| = 1$. Let S be a suitable set with $|S| < |N|$ and as defined before $S \subset N$. Let $a_S = 0_{|S|}$. Informally speaking the variables in S do not "exist" in P'. Let $a^T\lambda \leq \alpha$ be valid for P' (with $a, \lambda \in \mathbb{R}^N$ as defined before). Let $b \in \mathbb{R}^N$ with $b_{N\setminus S} = 0_{|N|-|S|}$. The nonzero elements in b (which are the elements of vector b_S) belong to the variables we want to lift in.
Lifting and separating means we look for a cut of the form $b^T\lambda + a^T\lambda \leq \alpha + \delta$ by applying the following algorithm:

Algorithm 39

1. *Select a "suitable" inequality $a^T\lambda \leq \alpha$ (perhaps a facet of P').*

2. *If the selected inequality is valid for P then goto 3. Else calculate*

$$\delta = max\{a^T\lambda^v - \alpha | \forall \lambda^v \in vert(P)\}.$$

   The inequality

$$a^T\lambda \leq \alpha + \delta \tag{6.1}$$

   is valid for P.

3. *Solve the following LP*

$$\begin{array}{rll} z^* = max & \bar{\lambda}^T b & \\ s.t. & (\lambda^v)^T b \leq \alpha + \delta - a^T\lambda^v & \forall \lambda^v \in vert(P). \end{array}$$

   Let $\bar{b}$ (from our assumption of course with $\bar{b}_{N\setminus S} = 0_{|N|-|S|}$ since only the elements of b_S are variable) be such that $\bar{b}^T\bar{\lambda} + a^T\bar{\lambda} - (\alpha+\delta) = z^*$. Then

$$\bar{b}^T\lambda + a^T\lambda \leq \alpha + \delta \tag{6.2}$$

   is valid for P.

We remark that 6.1 is valid because

$$\forall \lambda^v \in vert(P) : \alpha + \delta = \alpha + max\{a^T\lambda^v - \alpha | \forall \lambda^v \in vert(P)\} \geq \alpha + a^T\lambda^v - \alpha = a^T\lambda^v.$$

We give some facts for Algorithm 39:

(a) $\bar{b}^T\lambda + a^T\lambda \leq \alpha + \delta$ is valid for P.

   Proof. This proof is quite the same as the proof on page 83 in Chapter 5. We know that every feasible point of the polytop P can be combined as a convex combination of a suitable selection of its vertices $\lambda^v \in vert(P)$ (this is correct although P is not convex).

That is for $\lambda^* \in P$ *exist* nonnegative real numbers β_v for all $\lambda^v \in vert(P)$ and $\sum_{\lambda^v \in vert(P)} \beta_v = 1$ such that:

$$\lambda^* = \sum_{\lambda^v \in vert(P)} \beta_v \lambda^v.$$

We then calculate:

$$\begin{aligned}
&\bar{b}^T \lambda^* + a^T \lambda^* \\
&= \bar{b}^T \textstyle\sum_{\lambda^v \in vert(P)} \beta_v \lambda^v + a^T \sum_{\lambda^v \in vert(P)} \beta_v \lambda^v \\
&= \textstyle\sum_{\lambda^v \in vert(P)} \beta_v (\bar{b}^T \lambda^v) + \sum_{\lambda^v \in vert(P)} \beta_v (a^T \lambda^v) \\
&= \textstyle\sum_{\lambda^v \in vert(P)} \beta_v (\bar{b}^T \lambda^v + a^T \lambda^v) \\
&\leq \textstyle\sum_{\lambda^v \in vert(P)} \beta_v (\alpha + \delta) \\
&= (\alpha + \delta) \textstyle\sum_{\lambda^v \in vert(P)} \beta_v \\
&= \alpha + \delta.
\end{aligned}$$

Therefore $\bar{b}^T \lambda + a^T \lambda \leq \alpha + \delta$ is valid for P. □

(b) There exists a violated cut if and only if $z^* > 0$.

Proof. If $z^* > 0$ then due to (a) $\bar{b}^T \lambda + a^T \lambda \leq \alpha + \delta$ is such a cut. On the other hand, suppose $\tilde{b}^T \lambda + a^T \lambda \leq \alpha + \delta$ is a valid inequality violated by $\bar{\lambda}$ then $z^* \geq \tilde{b}^T \bar{\lambda} + a^T \lambda - (\alpha + \delta) > 0$. □

The problem for this algorithm is point 1. The selection of a suitable inequality can be very difficult and since the complexer discretisation can lead to a very complex polyhedron we do not know what inequality should be used. However here we give an example of a situation where this algorithm works:

We consider polyhedron P_Δ in the case described by Figure A.5 with

$$p^1_{out} = \begin{pmatrix} 25 \\ 18 \\ 26 \\ 24 \end{pmatrix}, p^2_{in} = \begin{pmatrix} 20 \\ 20 \\ 30 \\ 30 \end{pmatrix}$$

We want to separate the point

$$\begin{pmatrix} 0 \\ \frac{3}{4} \\ \frac{1}{4} \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

which is a vertex of the LP relaxation of P_Δ in the described situation but does not fulfil the triangle condition. From the polyhedron described by Figure A.2 we get from the situation

$$p^1_{out} = \begin{pmatrix} 25 \\ 18 \\ 24 \end{pmatrix}, p^2_{in} = \begin{pmatrix} 20 \\ 30 \\ 30 \end{pmatrix}$$

the inequality

$$\frac{1}{5}\lambda^1_4 + 2(\lambda^2_2 + \lambda^2_3) \leq 1.$$

This inequality is not valid for P_Δ in the case of Figure A.5. We calculate $\delta = \frac{1}{5}$.
According to step 3 of Algorithm 39 we solve the LP

$$\begin{array}{rl}
max & \frac{3}{4}\lambda^1_2 + \frac{1}{4}\lambda^1_3 + \lambda^2_2 \\
s.t. & \\
\frac{3}{5}\lambda^2_2 & \leq \frac{1}{5} \\
\frac{3}{5}\lambda^2_1 & \leq \frac{1}{5} \\
\frac{3}{5}\lambda^2_1 & \leq \frac{1}{5} \\
\lambda^1_3 + \frac{2}{5}\lambda^2_2 & \leq 0 \\
\lambda^1_3 + \frac{2}{5}\lambda^2_1 & \leq 0 \\
\lambda^1_3 + \frac{2}{5}\lambda^2_1 & \leq 0 \\
\frac{2}{3}\lambda^1_2 + \lambda^2_2 & \leq \frac{17}{15} \\
\frac{2}{3}\lambda^1_2 + \lambda^2_1 & \leq \frac{17}{15} \\
\frac{2}{7}\lambda^1_1 + \frac{5}{7}\lambda^1_2 + \lambda^2_2 & \leq \frac{6}{5} \\
\frac{2}{7}\lambda^1_1 + \frac{5}{7}\lambda^1_2 + \lambda^2_1 & \leq \frac{6}{5} \\
\lambda^1_1 + \frac{1}{2}\lambda^2_2 & \leq \frac{1}{5} \\
\lambda^1_1 + \frac{1}{2}\lambda^2_1 & \leq \frac{1}{5} \\
\lambda^1_1 + \frac{1}{2}\lambda^2_1 & \leq \frac{1}{5}
\end{array}$$

From Algorithm 39 (step 3) we get as new inequality

$$1.68\lambda^1_2 + \frac{1}{5}\lambda^1_4 + 2(\lambda^2_2 + \lambda^2_3) \leq \frac{6}{5}.$$

This inequality is valid and because of $1.68 \cdot 0.75 = 1.26 > 1.2$ it cuts off the LP solution.

Chapter 7

Implementation and Computational Results

7.1 Introduction

In this chapter we give an outline of some implementation details regarding a new Branch-and-Bound algorithm since the separation algorithms we have developed yet (see Chapter 5 and Chapter 6) cannot guarantee that the solution fulfils the set condition. With the additional Branch-and-Bound algorithm we explain in the next section we can be sure that we finally can calculate a solution which fulfils the set condition. At the end we give computational results for some test networks.

7.2 Branch-and-Bound for TTO

In this section we first derive a Branch-and-Bound algorithm which works for pipes as well as for compressors. After that we will give some ideas how to fasten this algorithm. We add that the presented branching algorithms only consider the binary variables which have been introduced for the approximation of nonlinearities. The switching variables of compressors, valves and control valves are not regarded here. For these variables there exist well known branching algorithms that are implemented in every state-of-the-art MIP solver.

7.2.1 A Branch-and-Bound Algorithm for Pipes and Compressors

Regrettably the several separation algorithms we have developed cannot guarantee that we are able to separate LP-solutions that do not fulfil the set conditions. Therefore we have to combine our separation algorithms with a suitable Branch-and-Bound rule. So let us consider Figure 7.1 which describes the well known triangulation of the pressure loss of a pipe. N^* with $|N^*| = n \in \mathbb{N}$ stands for the set of indices of all non-vanishing λ-variables. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the non-vanishing λ-variables of the solution of a LP-Relaxation of our problem. Assume some ordering of these variables.
Let N_i denote the neighbours of λ_i, that is the set of indices of λ-variables that are a corner of a triangle that contains λ_i as one of its three corners but not i itself.

Consider the following algorithm:

Algorithm 40

(1) For $i = 1$ To n Do

(2) If $N_i \cup \{i\} \not\supseteq N^*$ *goto* (4).

(3) End For

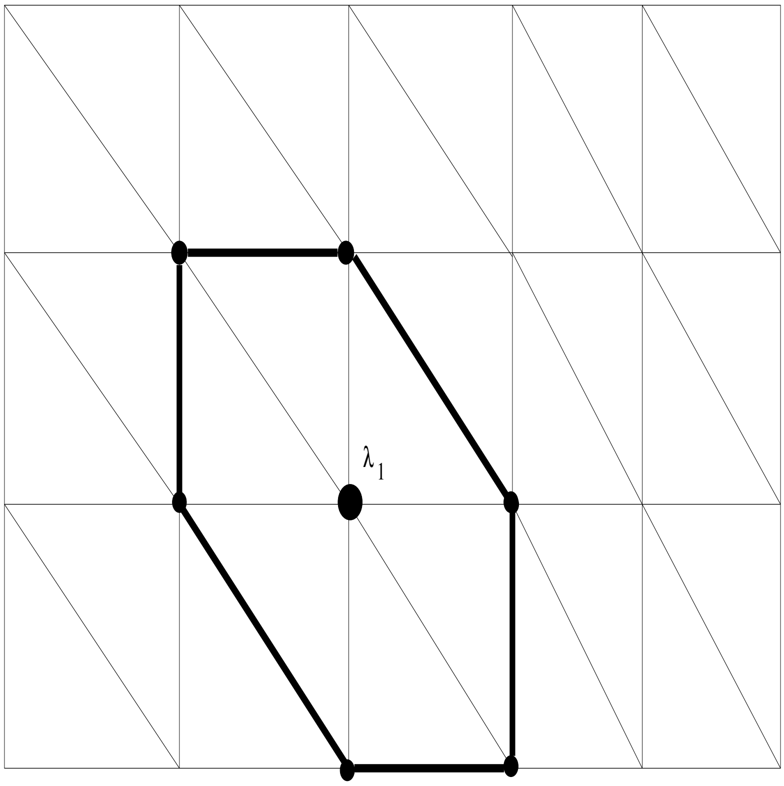

Figure 7.1: An example for branching (the neighbours of λ_1)

(4) Split the problem in the following way:

- *First subproblem:*

  Add the condition

$$\sum_{j \in N_i \cup \{i\}} \lambda_j = 1.$$

- *Second subproblem:*

  Add the condition

$$\lambda_i = 0.$$

(5) End

Lemma 41 *Algorithm 40 terminates with two branches in which the LP-solution is not feasible.*

Proof. We remark the following facts:
Obviously, in both subproblems the LP-solution is not feasible, since in the first case $N_i \cup \{i\} \subset N^*$ and in the second case $\bar{\lambda}_i > 0$.
One can show (after a somewhat long-winded and boring inspection of all possible cases - we will not specify these cases in detail) that after the third index latest the condition (2) in Algorithm 40 is satisfied and the algorithm terminates. □

In the same way (analogously as in the proof of Lemma 41) it is easy to see that in the case we use a triangulation in squares or rectangles Algorithm 40 terminates yet after the second index.

We just developed a Branch-and-Bound rule for pipes. If we know that a certain compressor cannot be switched off it is easy to develop a Branch-and-Bound rule in the case that the approximation of the fuel gas consumption is done by a triangulation of the domain of function f in cubes:
We define again N^* as the set of indices of all non-vanishing λ-variables and we again assume some ordering of these variables. In this case Algorithm 40 is completely the same. We only have to mention that we must adapt the set N_i of the neighbours of λ_i, that is now of course the set of indices of λ-variables that are a corner of a cube that contains λ_i as one of its eight corners but not i itself.

One can now easily (analogously as in the proof of Lemma 41) show that in this case Algorithm 40 terminates yet after the second index.

With the developed separation algorithms and the supplementary Branch-and-Bound rules we can guarantee that we are able to separate every LP-solution that does not fulfil the set condition without introducing binary variables!

At the end of this section we remark that we also can apply Algorithm 40 in the situation that we triangulate each cube in six similar tetrahedra in the case of a compressor (that cannot be switched off) in order to approximate the fuel gas consumption f (we have used this way for an effective implementation of the branching rule since there exist programming tools for such triangulations of cubes). We get a possible triangulation in the way that is pointed out in Figure 7.2: We intersect the cube along the three portly drawn lines. It it easy to see that the cube now resolves into 6 tetrahedra. The first tetrahedron has as its corners the points $1, 2, 4, 5$, the second $1, 3, 4, 5$, the third $2, 4, 5, 6$, the fourth $3, 4, 5, 7$, the fifth $4, 5, 6, 8$ and the sixth $4, 5, 7, 8$. It is clear how to understand the neighbours of a $\lambda-$variable in this situation: The set N_i of neighbours of λ_i is the set of indices of λ-variables that are a corner of a tetrahedron which contains λ_i as one of its four corners but not i itself.

One can show that also in this case Algorithm 40 terminates. The minimal index number after that the algorithm terminates is four.
We here give an alternative and more theoretical proof of the statement of Lemma 42:

Lemma 42 *Algorithm 40 in the case of tetrahedra also terminates with two branches in which the LP-solution is not feasible.*

Proof. We assume

$$N_l \cup \{l\} \supseteq N^* \qquad \forall l : \lambda_l > 0$$

which implies

$$\sum_{k \in N_l \cup \{l\}} \lambda_k = 1 \qquad \forall l : \lambda_l > 0$$

which means we assume that our algorithm does not terminate with two branches in which the LP-solution is not feasible.
Define $I = \{i \mid \lambda_i > 0\}$.
Let $E(I)$ be the set of all pairs (edges) of non-vanishing adjacent $\lambda-$variables in our triangulation.
Let us consider the graph $(I, E(I))$.
We assume there exist $i, j \in I$ with $(i, j) \notin E(I)$ that means i, j are in different tetrahedra.
Because of $\lambda_i > 0$ and $\lambda_j > 0$ and $(i, j) \notin E(I)$ we conclude

$$\sum_{k \in N_i \cup \{i\}} \lambda_k < 1$$

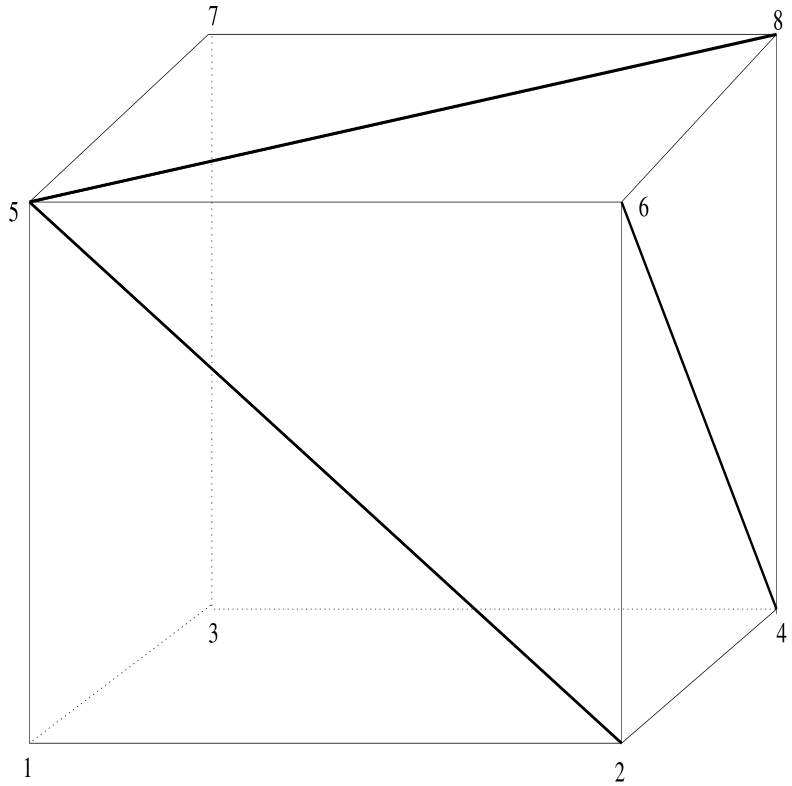

Figure 7.2: Triangulation of a cube in six similar tetrahedra

(since $\lambda_j > 0$ with $j \notin N_i$) and

$$\sum_{k \in N_j \cup \{j\}} \lambda_k < 1$$

(since $\lambda_i > 0$ with $i \notin N_j$). This is a contradiction to our first assumption and we get

$$(i, j) \in E(I) \qquad \forall i, j \in I$$

that means all positive $\lambda-$variables must be adjacent and therefore lie in a certain tetrahedron. We also see that the algorithm terminates after the fourth iteration. □

This proof also works in the case of triangles we examined first but the argument cannot be used in our second case of cubes since in a cube not all eight corners are connected. Remember that we already showed Lemma 42 in this case.

We shortly come back to the problem of ordering the variables.
There are two easy ideas to order the $\lambda-$variables. At first we can order the variables such that holds

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n > 0$$

or such that holds

$$0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n.$$

We notice as a result of our test calculations that the branching idea we discussed usually produces shorter computation times in connection with the described separation algorithm for pipes if we use the order $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.

7.2.2 Additions to the described Branch-and-Bound Algorithm

The advantage of the described Branch-and-Bound algorithm is that we can guarantee finally to calculate a solution that fulfils the set condition. We give some further ideas which can perhaps sometimes be helpful to fasten our branching algorithm:

Since we approximate the pressure loss function of a pipe by a set Λ_p of two-dimensional grid points we argue as follows:
It is sufficient to concentrate on triangulations which have the form shown in Figure 7.3. To generalise the following arguments is quite easy but this is not necessary to do for our calculations. Let $k \in \mathbb{N}$ be the number of values in which we divided the interval between the minimal and maximal pressure at the beginning of a pipe. Now we define a set $N_{p_{in}} = \{p^i_{in,grid} \mid i = 1, \ldots, k\}$ where the values $p^i_{in,grid}$ are the "possible" pressure values at the grid points (w.l.o.g. in ascending order, see as an example 5.3). Now we define for $i = 1, \ldots, k$ sets $N^i_{p_{in}} = \{j \mid p^j_{in} = p^i_{in,grid}\}$ (see the notation in Chapter 4).
Define now for $1 < i < k$

$$s^i_a = \sum_{j \in N^i_{p_{in}}} \lambda^j.$$

Additionally we define for $i > 1$

$$s^i_l = \sum_{j \in \bigcup_{l=1}^{i-1} N^l_{p_{in}}} \lambda^j$$

and for $i < k$

$$s^i_r = \sum_{j \in \bigcup_{l=i+1}^{k} N^l_{p_{in}}} \lambda^j.$$

Here subindex l indicates the sum of the $\lambda-$variables on the "left" side, r the sum of the $\lambda-$variables on the "right" side and subindex a indicates the set of the "actual" $\lambda-$variables we just have selected for branching, see Figure 7.3.
The principle idea is to select a suitable number i such that we can split the problem into two subproblems (here we cannot ensure to calculate two branches in which the LP-solution is not feasible):

- First subproblem:

  Add the condition

$$s^i_l + s^i_a = 1. \tag{7.1}$$

- Second subproblem:

  Add the condition

$$s^i_a + s^i_r = 1. \tag{7.2}$$

A prudential criterium for a selection of $i = 2, \ldots, k-1$ is to require

$$max_{i=2,\ldots,k-1}[(1 - (s^i_l + s^i_a)) + (1 - (s^i_a + s^i_r))]$$

The number $1 - (s^i_l + s^i_a)$ is a measure of the "value we cut off" in the first subproblem and the number $1 - (s^i_a + s^i_r)$ is a measure of the "value we cut off" in the second subproblem. Our condition means that we want to maximise the sum of these two values (without any weighting of the two sums). We calculate $max_{i=2,\ldots,k-1}[(1 - (s^i_l + s^i_a)) + (1 - (s^i_a + s^i_r))] = max_{i=2,\ldots,k-1}[2 - 2s^i_a - s^i_l - s^i_r] =$

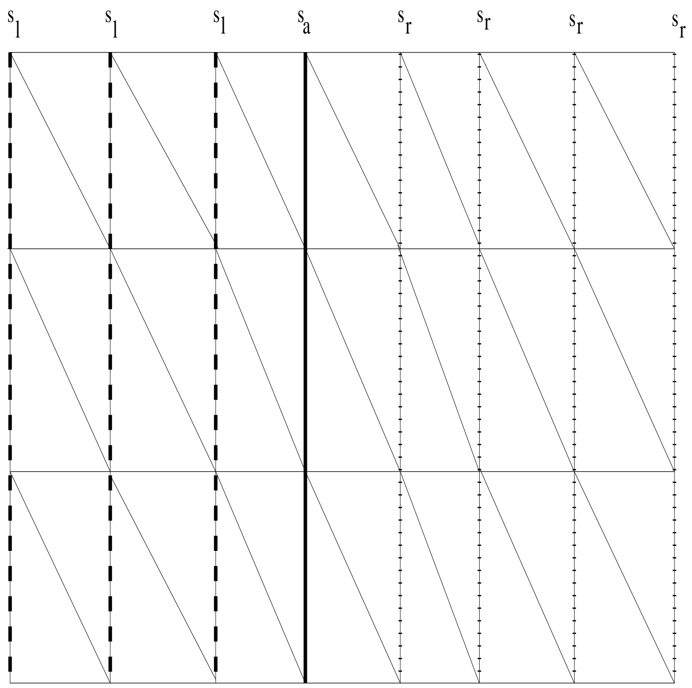

Figure 7.3: An example for horizontal/vertical branching

$max_{i=2,...,k-1}[1 - s_a^i]$ since we know $s_l^i + s_a^i + s_r^i = 1 \; \forall i = 2, \dots, k-1$. Because of $0 \leq s_a^i \leq 1 \; \forall i = 2, \dots, k-1$ we get

$$max_{i=2,...,k-1}[(1 - (s_l^i + s_a^i)) + (1 - (s_a^i + s_r^i))] = min_{i=2,...,k-1} s_a^i.$$

Considering a certain pipe we can do the same procedure with the gas flow in this pipe. That means we define a set $N_q = \{q_{grid}^i \mid i = 1, \dots, k\}$ where the values q_{grid}^i are the "possible" pressure values at the grid points (in ascending order). Now we define sets $N_q^i = \{j \mid q^j = q_{grid}^i\}$.
The rest is quite the same as we described it for the pressure p_{in} at the beginning of the pipe. We leave the formal description of the complete algorithm in this case.

We add another simple idea: In the case that the difference between the linearised pressure at the end of the pipe (that means this value in the LP-solution) and the exact pressure at the calculated point $(p_{in}, q)^T$ of the LP-solution is very small (that means smaller as a value $\epsilon > 0$) we can do without separating or branching on the calculated LP-solution. The same we can do with a compressor.

7.2.3 Combining the Ideas for Branching

Combining our ideas we get the following branching algorithm for a pipe (for one new branch node):

Algorithm 43

(1) Check, whether the calculated LP-solution does fulfil the set condition. If it does, goto (5).

(2) If there are two branches concerning p_{in} in which the LP-solution is not feasible add 7.1 and 7.2 in this case and goto (5).

(3) If there are two branches concerning q in which the LP-solution is not feasible add 7.1 and 7.2 in this case and goto (5).

(4) Apply Algorithm 40. We have already shown that there are now two branches in which the LP-solution definitely is not feasible.

(5) End.

We remark that in some cases Algorithm 43 leads to shorter calculation times than Algorithm 40 (the maximal number of branches for Algorithm 43 then is lower than for Algorithm 40).
The extension of Algorithm 43 for a compressor now is obvious. We only need to remember that for a pipe we linearised a nonlinear function of the form $p_{out} = p_{out}(p_{in}, q)$. Since the fuel gas consumption f is a function of the form $f = f(p_{in}, p_{out}, q)$ we can easily generalise Algorithm 43 to Algorithm 44. We pass on developing a formal description of the details of the complete algorithm since the idea is obvious.

Algorithm 44

(1) Check, whether the calculated LP-solution does fulfil the set condition. If it does, goto (6).

(2) If there are two branches concerning p_{in} in which the LP-solution is not feasible add 7.1 and 7.2 for the compressor and goto (6).

(3) If there are two branches concerning p_{out} in which the LP-solution is not feasible add 7.1 and 7.2 and goto (6).

(4) If there are two branches concerning q in which the LP-solution is not feasible add 7.1 and 7.2 and goto (6).

(5) Apply Algorithm 40 in the case we described in Figure 7.2. No later than now there are two branches in which the LP-solution is not feasible.

(6) End.

Summarising all ideas we illuminated here and in Chapter 5 we see that we have developed a complete Branch-and-Cut Algorithm for the binary variables introduced for the approximation of nonlinearities of the Mixed Integer Problem we described in Chapter 4.

7.3 Some Computational Results

In this section we finally give calculations for some gas networks. We proceed as follows:
We start with the small test network we discussed in Chapter 4. Here we show in dependency from the accuracy of the discretisation the computational progress when using our separation algorithm (for pipes) instead of using binary variables in the traditional formulation we exploited in Chapter 4. In this calculation the branching routine is not used because branching can become quite inefficient for fine discretisations. But the table shows us that using the cuts we get quite good solutions for this model and branching is not necessary.

After that we give some calculations for a somewhat complexer gas network. Here we work with a constant discretisation and use all developed algorithms. We show the dependency of the solution time from the input data and we give three tables of the same examples: in the first table we calculate with separation **and** branching algorithms, in the second table we only use separation algorithms (that means we solve the root node) and after that in the third table we only use branching and do not calculate cuts. For every table we stop if the difference between the linearised pressure and the exact pressure at the calculated point $(p_{in}, q)^T$ of the LP-solution for every node is smaller than some value ϵ (see the previous section). Finally we give an example of a simplified real gas network and show that using crude

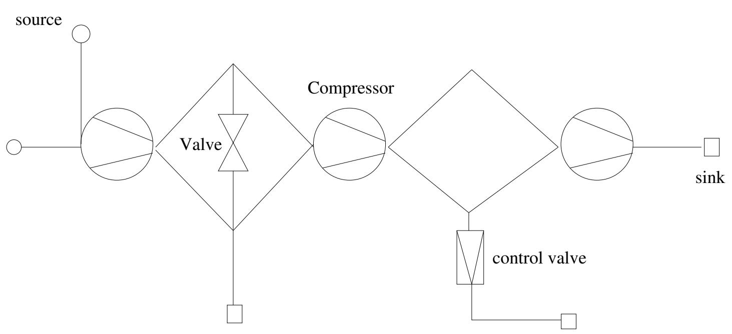

Figure 7.4: A simple gas network

compressors ($\square, y$)			pipes ($\triangle$)		Solution			
$p_{in,C}$	$p_{out,C}$	q_C	$p_{in,P}$	q_P	CPLEX cuts	User cuts	Opt val	sec
3	3	7	4	10	29	0	9.39	3.07
3	3	7	4*	10*	6	10	9.36	0.79
3	3	7	8	20	28	0	9.16	295.9
3	3	7	8*	20*	7	204	9.15	23.09

discretisations we are also able to give a reasonable solution. A suitable idea seems to be after that to calculate the problem now again with a nonlinear optimisation tool since now the binary variables are fixed and since the global optimum should be found in a neighbourhood of the calculated solution. As LP-solver CPLEX 8.0 was used. The calculations were done on a 1 GHz Pentium III processor with 1 GB main memory.

7.3.1 Comparison of Binary Variables and Cuts

We have tested our implementation of the algorithm for the polyhedron $P_{\triangle}$ as it was described on page 57 for a gas network which consists of three compressors and ten pipes. This gas network is shown in Figure 7.4. In our first formulation of the model we used the traditional way of the introduction of binary variables for modelling piecewise linear functions. That is we introduce for each triangle $i \in \Lambda$ a binary variable y_i and model the fact that all positive $\lambda-$variables must belong to the same triangle. The computational results for this model are indicated by y in the table above. The table shows our experiences of the computational progress when incorporating the polynomial separation algorithm instead of binary variables (here the compressors are still formulated with binary variables but the pipes are using already the cuts obtained from the separation algorithm).

$p_{in,C}$ is the number of grid points used for the pressure at the beginning of a compressor. $p_{out,C}$ analogously describes the number of grid points for the pressure at the end of a compressor. q_C is the number of grid points for the gas flow of the compressor. $p_{in,P}$ is the number of grid points used for pressure at the beginning of a pipe and q_P means the number of grid points for the gas flow in the pipe. In the rows in which the number of user cuts (constructed by the separation algorithm) is zero the problem was calculated by the formulation with binary variables. We see that the use of cuts constructed by the separation algorithm reduces the calculation time about factor 10. Column 8 compares all solution values of this model for the calculation with binary variables and with the cuts. The differences are negligible. Only using cuts produces good approximations to the optimal solution.
Let us give a short comment about our implementation: The LP-relaxations are calculated with CPLEX (we used CPLEX 7.0, for the branching rules CPLEX 8.0 is needed). We are working with the CPLEX cutcallback functions. Callbacks may be called repeatedly at various points during an optimisation.

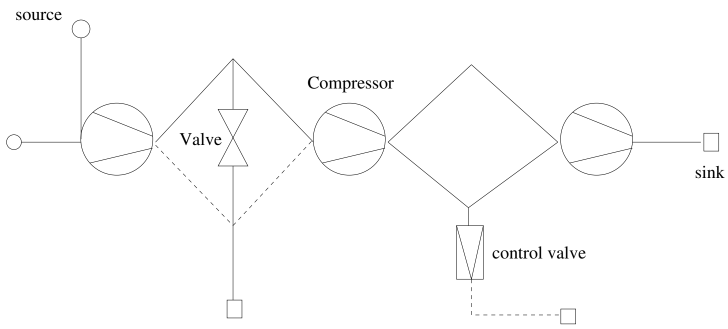

Figure 7.5: The test model before the separation algorithm for pipes

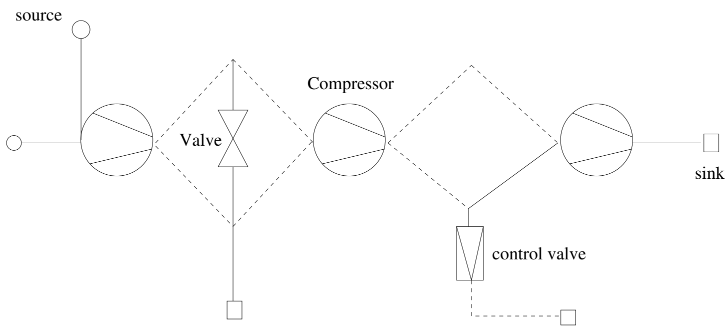

Figure 7.6: The test model after the separation algorithm for pipes

Here we are looking in each LP-iteration for a cut we have calculated by the separation algorithm. This cut is added to the LP-relaxation of the problem.

Figure 7.5 shows the situation before using the constructed cuts. The solid lined pipes do not fulfil the triangle (set) conditions whereas the dotted pipes do. In Figure 7.6 we can see the situation after the use of the separation algorithm. We see that in Figure 7.6 still one pipe does not fulfil the triangle condition. The reason for this is that the polyhedron P_Δ (in general the polyhedron P) is not convex. So in some cases it can be possible that the solution to be cut off lies in the interior of the convex hull of the polyhedron P but not in P itself. Such points cannot be cut off by a valid inequality. Here the branching algorithm of Chapter 7 ensures that we can cut off such points that do not fulfil the set conditions. But in fact it is not always necessary to fulfil the set conditions for every segment since the solution values often are good enough and as we stated we have the differences under control.

These first test calculations we presented showed us that the theoretical knowledge of the vertices and the separation algorithm give us the possibility to extend our Branch-and-Cut algorithm to complexer gas networks in order to reduce the solution time significantly.

7.3.2 Using Cuts and Branching

Figure A.7 (see page 154) shows a modified and little heightened version of the test model we have already studied. Let us come to the test model in Figure A.7 now. The complete test data can be found in the Appendix.

Solving the problem with our implemented Branch-and-Cut algorithm the solution time is 18.6 seconds. The separation algorithm for polyhedron P_Δ (see Chapter 5) produced 268 cuts. The branching routine was needed for 14 times. In this case we use the separation algorithm in all cases and the branching rule with $\epsilon = 0.05$ (ϵ is understood in the sense described above).

The following table gives an impression of the different behaviour of the model while changing exactly *one value for a single node or segment* (while the same discretisations of pipes and compressors are used; we always calculated with $p_{in,C} = p_{out,C} = 4, q_C = 10$ and $p_{in,P} = 4, q_P = 10$, see page 52):

Node/Segment	Value	Solution time(sec.)	Cuts	Branching
C02	$\eta_{ad} = 0.75$	11.68	187	1
L16	$L = 70000.0$	7.56	160	1
L16	$k = 0.00005$	19.14	268	14
C03	$\eta_{ad} = 0.65$	14.12	194	2
Q03	$q_{min} = 0.0$	21.41	259	22
A01	$q_{min} = 490.0$	18.7	183	16
A01	$q_{min} = 495.0$	32.88	245	88
A01	$q_{min} = 499.0$	58.76	401	182
A01	$q_{min} = 500.0$	110.26	396	286
A01	$q_{min} = 501.0$	24.48	235	93
A01	$q_{min} = 505.0$	69.77	359	198
A01	$q_{min} = 510.0$	107.93	446	290
A02	$q_{min} = 100.0$	11.76	143	1
A03	$q_{min} = 100.0$	4.8	86	1
A04	$q_{max} = 600.0$	15.75	215	2
A04	$p_{min} = 45.0$	12.27	180	1
A05	$q_{min} = 500.0$	22.58	315	26
A05	$p_{min} = 45.0$	11.39	193	1
A06	$p_{min} = 45.0$	10.01	185	1

As a consequence we see that small and unimposing changings of model parameters can lead to significant changes of the solution of the model. This is a great problem when solving Mixed Integer Linear Problems with Branch-and-Cut algorithms.

7.3.3 Using only Cuts

Let us give some further calculations for this model:
Here we calculate the solution of the root node.
The solution time of this problem is 2.670 seconds and we calculate 62 cuts. The difference to the solution with branching is only 1% which seems to be justifiable according to the shorter calculation time.
Here we give the calculations when changing one value for a single segment:

Node/Segment	Value	Solution time(sec.)	Cuts	Branching
C02	$\eta_{ad} = 0.75$	2.78	76	0
L16	$L = 70000.0$	3.19	124	0
L16	$k = 0.00005$	2.67	62	0
C03	$\eta_{ad} = 0.65$	2.56	61	0
Q03	$q_{min} = 0.0$	2.07	64	0
A01	$q_{min} = 490.0$	3.79	100	0
A01	$q_{min} = 495.0$	3.35	101	0
A01	$q_{min} = 499.0$	3.58	94	0
A01	$q_{min} = 500.0$	4.08	128	0
A01	$q_{min} = 501.0$	4.4	110	0
A01	$q_{min} = 505.0$	3.2	98	0
A01	$q_{min} = 510.0$	2.46	87	0
A02	$q_{min} = 100.0$	2.25	49	0
A03	$q_{min} = 100.0$	2.98	63	0
A04	$q_{max} = 600.0$	2.76	62	0
A04	$p_{min} = 45.0$	5.16	144	0
A05	$q_{min} = 500.0$	3.24	98	0
A05	$p_{min} = 45.0$	4.55	98	0
A06	$p_{min} = 45.0$	4.92	133	0

7.3.4 Using only Branching

In the next table we calculate the same situations with $\epsilon = 0.01$ for the branching rule. The separation algorithm is not used.
The calculation time of the initial problem is about 12.75 seconds. The branching routine is used 960 times.

Node/Segment	Value	Solution time(sec.)	Cuts	Branching
C02	$\eta_{ad} = 0.75$	6.13	0	473
L16	$L = 70000.0$	3.95	0	282
L16	$k = 0.00005$	11.16	0	960
C03	$\eta_{ad} = 0.65$	9.42	0	855
Q03	$q_{min} = 0.0$	5.45	0	489
A01	$q_{min} = 490.0$	2.58	0	46
A01	$q_{min} = 495.0$	1.31	0	6
A01	$q_{min} = 499.0$	1.5	0	4
A01	$q_{min} = 500.0$	1.92	0	28
A01	$q_{min} = 501.0$	1.06	0	4
A01	$q_{min} = 505.0$	1.71	0	6
A01	$q_{min} = 510.0$	1.72	0	12
A02	$q_{min} = 100.0$	12.14	0	746
A03	$q_{min} = 100.0$	13.26	0	658
A04	$q_{max} = 600.0$	7.79	0	703
A04	$p_{min} = 45.0$	9.19	0	766
A05	$q_{min} = 500.0$	22.36	0	988
A05	$p_{min} = 45.0$	11.08	0	960
A06	$p_{min} = 45.0$	12.71	0	1090

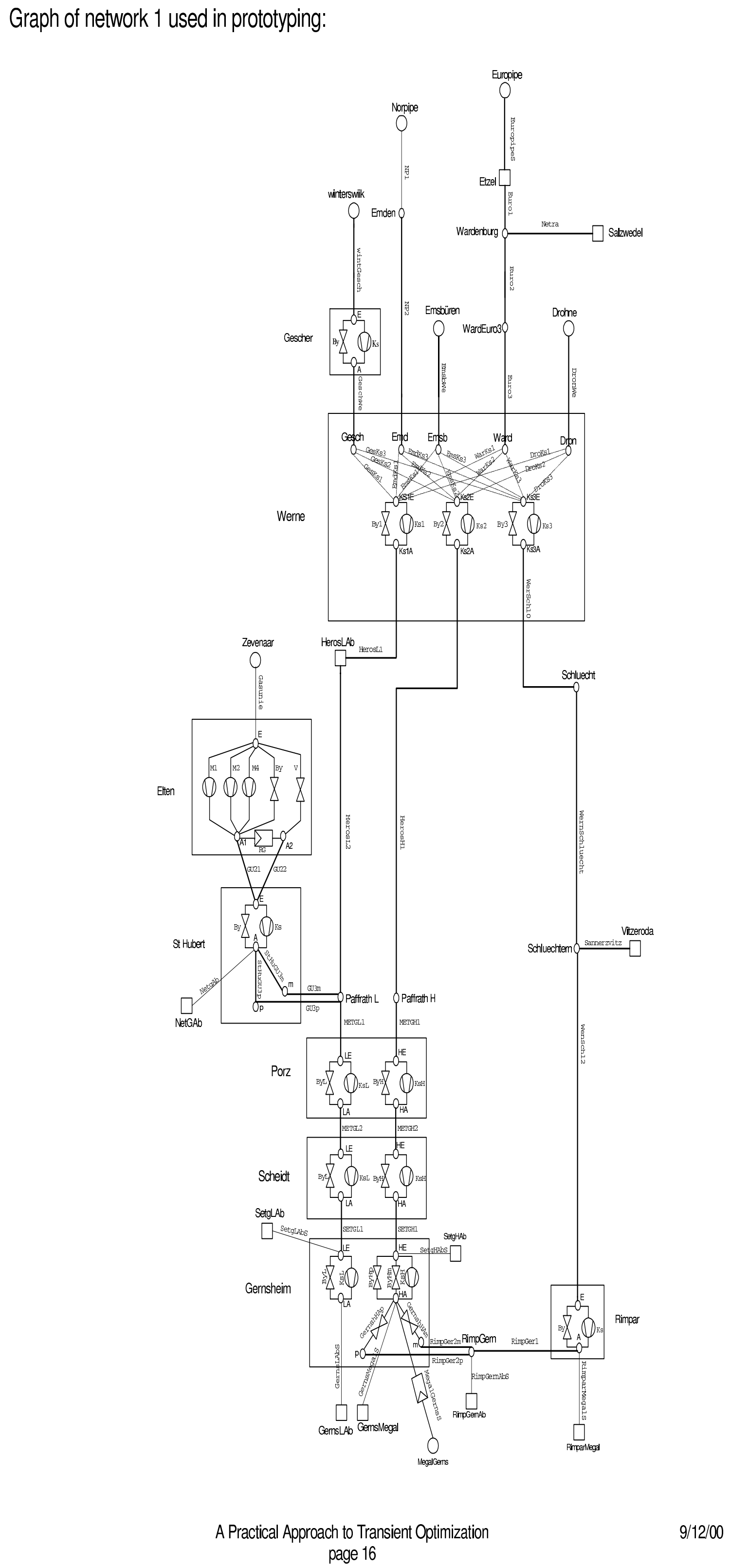

Figure 7.7: A huge test model, see [21]

ϵ	$sec.$	$Cuts$	$Branching$
0.05	7.5	0	3
0.05	10.3	224	0
0.01	62.5	0	1385
0.01	34.3	165	439
0.001	1072.1	0	35656
0.001	543.2	163	15522
0.0001	1425.7	0	49534
0.0001	619.2	163	15036

Table 7.1: Solution quality depending on accuracy parameter

7.3.5 A Further Example and Concluding Remarks

Figure 7.7 shows a simplification of the complete gas network of the German Ruhrgas AG. First test calculations showed that the developed algorithms give us the possibility to calculate the stationary case of this gas network in a justifiable time. A big problem is that the solution time of this model extremely depends on the input data. Here more research work in order to get rid of this problem is necessary.

One important positive result of our test calculations is that for not too small ϵ the implemented branching algorithm only rarely has to be used and that the most important improvements of the LP-solutions are achieved by the separation algorithm.

A further analysis of the test calculations shows us that the difference of the objective value when using separation and branching algorithms or only the separation or the branching algorithm is usually small.

The following calculations show that the constructed cuts usually lead faster to a solution than the only use of the branching algorithm. Here we calculated the second test example for different values of ϵ first only with the separation and second only with the branching algorithm. ϵ is understood in the sense described above. The table $Cuts$ shows how often the separation algorithm is used where the table $Branching$ shows how often the branching heuristic was needed.

ϵ	$sec.$	$Cuts$	$Branching$
0.05	7.5	0	3
0.05	22.3	6995	0
0.01	58.46	0	1385
0.01	28.27	10633	0
0.001	1072.1	0	35656
0.001	23.58	8587	0
0.0001	1425.7	0	49534
0.0001	23.35	8499	0

In the table above in the rows where no branching was used we only solved the root node. This can be done in a justifiable time. For the sake of completeness we give the same table (see Table 7.1) calculated with a so called cut-and-branch algorithm. The difference to the branch-and-cut idea is that we here only use cuts in the root node and then proceed by branching. Clearly the calculation time now increases but we see that using cuts is better than only using branching.

Taking our experiences into account the following strategy may be useful: The combination of all developed algorithms cannot in all situations guarantee to calculate the optimum in a short time. So, if we want to calculate the optimal solution for a more complex or real-world situation we first should try either the separation or for a not too small ϵ the branching algorithm. After that a use of a quick

Compressor			Pipe		Solution, $\epsilon = 0.05$		Solution, $\epsilon = 0.01$	
$p_{in,C}$	$p_{out,C}$	q_C	$p_{in,P}$	q_P	Opt. val.	sec	Opt. val.	sec.
3	3	3	3	7	8.4514	0.32	8.4514	0.34
3	3	7	3	9	8.6237	0.88	8.6396	1.35
3	3	7	4	8	7.8031	1.08	7.8277	11.5
3	3	7	4	10	7.7140	1.8	7.7426	10.33
4	4	10	4	10	7.7734	10.3	7.7880	34.3
4	4	10	4	15	7.6817	12.86	7.6914	10.58
4	4	10	4	20	7.6585	26.44	7.6614	15.18
4	4	10	5	10	7.3819	11.06	7.4709	299.31
4	4	10	5	12	7.3158	20.39	7.3280	27.01

Table 7.2: Solution qualities depending on grid sizes

nonlinear optimisation tool in the neighbourhood of the calculated solution could be helpful in order to calculate the real (global) optimum (remember that we know the values of switching variables now so that we have a typical nonlinear optimisation problem with a good starting value).

Nevertheless, already these few test calculations show us that our separation algorithm and thus our understanding of the properties of polyhedron P from Chapter 5 is subtotal.

The following table illustrates that the solution time usually extremely increases in dependency of the used grids (the table considers the second test model in a special situation with different grids, we used branch-and-cut with $\epsilon = 0.05$):

Comprerssor			Pipe		Solution	
$p_{in,C}$	$p_{out,C}$	q_C	$p_{in,P}$	q_P	Opt. val	sec
3	3	3	3	7	8.7210	0.54
3	3	7	3	9	8.6237	1.19
3	3	7	4	8	7.8494	22.6
3	3	7	4	10	7.7153	2.77
4	4	10	4	10	7.7752	18.64
4	4	10	4	12	7.7178	22.65
4	4	10	4	15	7.6898	38.14
4	4	10	4	20	7.6506	87.28
4	4	10	5	10	7.4393	447.73
4	4	10	5	12	7.3235	877.27

Table 7.2 shows the previous calculations with $\epsilon = 0.05$ and $\epsilon = 0.01$ using cut-and-branch.
At the end let us give Table 7.3 with selected calculations for the three test networks (here we used the same grid for every segment and cut-and-branch with $\epsilon = 0.05$ and $\epsilon = 0.01$). Cut-and-branch seems to conclude to good solutions even for bigger gas networks (here branch-and-cut usually will be more time consuming). Generally we notice that the calculation time for our MIP-Problems is usually quite longer than the calculation time for a nonlinear solver (with fixed binary variables) but until now to the best of our knowledge no suitable nonlinear solver *with binary variables* (that would be able to solve the problems we studied here) has been developed and so our idea to approximate the nonlinear problem by a linear MIP-Problem seems to be a prudential approach. However, our results show that a solution of the problem in its whole difficulty will require further research.

Pipes	Compressors	Total length of pipes	Time ($\epsilon = 0.05$)	Time ($\epsilon = 0.01$)
11	3	920	1.23 sec	1.99 sec
20	3	1200	1.17 sec	9.89 sec
31	15	2200	11.5 sec	104.4 sec

Table 7.3: Cut-and-Branch for different gas networks

As a consequence we see the following facts:

- We developed a flexible model regarding to the complexity of the used test network.
- The quality of approximation is algorithmically manageable.
- The calculation times increase moderately in dependency of the used test model but little differences of test data and the used grids can lead to quite huge differences in calculation time (compare the last two tables).

Concluding Remarks

At the end of such time-consuming work everyone should be harsh on himself:

The aim of the work presented in this thesis was to develop and solve a mixed integer model for the optimisation of gas networks. How far we have been able to achieve this aim?

We have been able to develop a model for the stationary case and a separation and branch-and-bound algorithm in order to get rid off binary variables and to fasten our calculations. But we have also seen that the force of our algorithms depends on the used discretisation. Using very fine approximations particularly the branching algorithms is very time-consuming. So we need to restrict ourselves to certain approximations of the problem. Under these conditions our algorithms seem to work pretty well since the quality of approximation is algorithmically manageable and in our test examples we got quite good solutions. However, it seems to be very complicated to approximate such a complex nonlinear optimisation problem with binary variables by a MILP; but we have already mentioned that there is still no optimisation tool for such complex nonlinear optimisation problems.

It is a little unfortunate that the model we developed only considers the stationary case. But our implementation and first tests of the transient model showed that from the beginning there was no prospect of solving the full problem in an acceptable time range.

Comparing our ideas and algorithms - although they must be called premature - with the attempts that already have been done on the problem of the Transient Technical Optimisation we risk the bold statement that our ideas and attempts after some further analysis and development could make a reasonable contribution to this very complex problem. For our justification we add that our ideas are quite general and can be used for other problems, too.

The necessary further research work especially the analysis of the complete time depending model will be part of the work on another graduation thesis in the research group of the tutor of this thesis. Time will show if this work will come to a good end - and as Shakespeare said:

All's Well That Ends Well.

Appendix A

Facets or Valid Inequalities for small Triangulations of $P_\triangle$

A.1 General Remarks

In our considerations on developing a lifting algorithm our focus was on the equality of the pressure at the end of a pipe and the beginning of the following pipe. In every gas network this case is very common. In our discussion of the facets and valid inequalities of small triangulations we did not consider the gas flow. In the case of one ingoing and one outgoing pipe often also the facets or valid inequalities calculated here may be used for lifting which is easy to see since we formally have the same problem in this situation. The problem when we are considering the gas flow is that because of the first law of Kirchhoff the situation in the case of several ingoing and several outgoing pipes is much more difficult because here we have to consider the equality of sums of gas flows. We tried to calculate facets and valid inequalities for small cases considering the Kirchhoff law in more complicated situations as one ingoing and one outgoing pipe. Unfortunately the complexity is to big to find general formulas. If we want to model gas flow preservation we have to go back to our first algorithm for the calculation of the vertices of such polytopes. Then we can use the separation algorithm we have presented in the Chapter 5.

A.2 Facets and Valid Inequalities

This part of the appendix belongs to Chapter 6. We consider the already known polyhedron $P_\triangle$ in the following form:

$$P_\triangle = (\{\tbinom{\lambda^1}{\lambda^2}| \quad \begin{array}{c} \sum_j \lambda_j^1 = 1 \\ \sum_i \lambda_i^2 = 1 \\ \sum_j p_{out,j}^1 \lambda_j^1 = \sum_i p_{in,i}^2 \lambda_i^2 \\ \lambda^1, \lambda^2 \geq 0 \\ \lambda^1, \lambda^2 \text{satisfy the triangle condition}\}) \end{array}$$

with

$$\lambda^1 = \begin{pmatrix} \lambda_1^1 \\ \lambda_2^1 \\ \lambda_3^1 \end{pmatrix}, \lambda^2 = \begin{pmatrix} \lambda_1^2 \\ \lambda_2^2 \\ \lambda_3^2 \end{pmatrix}, p_{out}^1 = \begin{pmatrix} p_{out,1}^1 \\ p_{out,2}^1 \\ p_{out,3}^1 \end{pmatrix}, p_{in}^2 = \begin{pmatrix} p_{in,1}^2 \\ p_{in,2}^2 \\ p_{in,3}^2 \end{pmatrix}$$

where $p_{out}^1, p_{in}^2 \in \mathbb{R}^3$ are the two vectors of pressure values at the grid points described by the λ-variables λ^1, λ^2.

In the following tables we first specify the vertices of the studied cases and after that we give the **non-trivial** facets (of the convex hull) of $P_\triangle$ (under a nontrivial facet we understand the facets that are not part of the original description of the polyhedron). The values $\alpha, \beta, \gamma, \delta, \epsilon, \zeta$ depend on the pressure

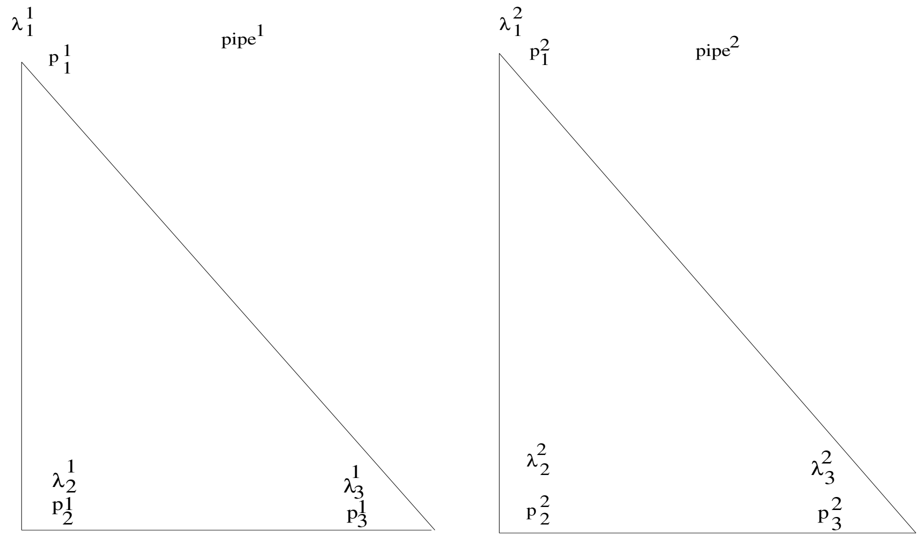

Figure A.1: Building vertices and facets of the polyhedron P_Δ

values (p^1_{out}, p^2_{in}) and can be calculated very easy in the way of Example 10 in Chapter 5. We note that the inequalities can also be interpreted independent from the pressure values since our calculations only use the structure of the vertices.

As an example: Let in the table (w.l.o.g. $j < k$) for a vertex $\lambda^1_i = 1, \lambda^2_j = \alpha, \lambda^2_k = 1 - \alpha$. Then we know from Algorithm 2 in Chapter 5 for the calculation of the vertices of P_Δ:

$$\alpha = \frac{p^1_i - p^2_k}{p^2_j - p^2_k}.$$

Analogously let in the table (w.l.o.g. $j < k$) for a vertex $\lambda^1_i = \alpha, \lambda^1_j = 1 - \alpha, \lambda^2_k = 1$. Then we know from our algorithm for the calculation of the vertices of P_Δ:

$$\alpha = \frac{p^2_k - p^1_j}{p^1_i - p^1_j}.$$

We remark that we did not concentrate on finding the complete description of the polyhedron.

We first consider the situation which is described in Figure A.1.
General Assumption: $p^2_2 = p^2_3, p^2_1 < p^2_2, p^1_1 < p^1_2, p^1_2 > p^1_3$.

Case 1: $p^1_1 > p^2_2, p^1_3 < p^2_1$

Ingoing pipe			Outgoing pipe		
λ^1_1	λ^1_2	λ^1_3	λ^2_1	λ^2_2	λ^2_3
0	α	$1-\alpha$	1	0	0
0	β	$1-\beta$	0	0	1
0	β	$1-\beta$	0	1	0
γ	0	$1-\gamma$	1	0	0
δ	0	$1-\delta$	0	0	1
δ	0	$1-\delta$	0	1	0

$$\begin{aligned} -\frac{1}{1-\gamma}\lambda_3^1 - \frac{\delta-\gamma}{1-\gamma}(\lambda_2^2+\lambda_3^2) &\leq -1 \\ \frac{1}{1-\alpha}\lambda_3^1 + \frac{\alpha-\beta}{\alpha-1}(\lambda_2^2+\lambda_3^2) &\leq 1 \end{aligned}$$

Case 2: $p_1^2 < p_1^1 < p_2^2, p_2^1 < p_2^2, p_3^1 < p_1^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	1	0	0
0	1	0	β	0	$1-\beta$
0	1	0	β	$1-\beta$	0
γ	0	$1-\gamma$	1	0	0
1	0	0	δ	0	$1-\delta$
1	0	0	δ	$1-\delta$	0

$$\begin{aligned} -\frac{1}{1-\gamma}\lambda_3^1 - \frac{1}{1-\delta}(\lambda_2^2+\lambda_3^2) &\leq -1 \\ \frac{1}{1-\alpha}\lambda_3^1 + \frac{1}{1-\beta}(\lambda_2^2+\lambda_3^2) &\leq 1 \end{aligned}$$

Case 3: $p_1^2 < p_1^1 < p_2^2, p_2^1 > p_2^2, p_3^1 < p_1^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	1	0	0
0	β	$1-\beta$	0	0	1
0	β	$1-\beta$	0	1	0
γ	0	$1-\gamma$	1	0	0
δ	$1-\delta$	0	0	0	1
δ	$1-\delta$	0	0	1	0
1	0	0	ϵ	0	$1-\epsilon$
1	0	0	ϵ	$1-\epsilon$	0

$$\begin{aligned} \frac{1}{1-\alpha}\lambda_3^1 + \frac{\alpha-\beta}{\alpha-1}(\lambda_2^2+\lambda_3^2) &\leq 1 \\ -\frac{1}{1-\gamma}\lambda_3^1 - \frac{1}{1-\epsilon}(\lambda_2^2+\lambda_3^2) &\leq -1 \end{aligned}$$

Case 4: $p_1^1 < p_1^2, p_1^2 < p_2^1 < p_2^2, p_3^1 < p_1^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	1	0	0
0	1	0	β	0	$1-\beta$
0	1	0	β	$1-\beta$	0
γ	$1-\gamma$	0	1	0	0

$$\frac{1}{1-\alpha}\lambda_3^1 + \frac{1}{1-\beta}(\lambda_2^2 + \lambda_3^2) \quad \leq 1$$

Case 5: $p_1^1 < p_1^2, p_2^1 > p_2^2, p_3^1 < p_1^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	1	0	0
0	β	$1-\beta$	0	0	1
0	β	$1-\beta$	0	1	0
γ	$1-\gamma$	0	0	0	1
γ	$1-\gamma$	0	0	1	0
δ	$1-\delta$	0	1	0	0

$$\frac{1}{1-\alpha}\lambda_3^1 + \frac{\alpha-\beta}{\alpha-1}(\lambda_2^2 + \lambda_3^2) \quad \leq 1$$

Case 6: $p_1^1 > p_2^2, p_1^2 < p_3^1 < p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	α	0	$1-\alpha$
0	0	1	α	$1-\alpha$	0
0	β	$1-\beta$	0	0	1
0	β	$1-\beta$	0	1	0
γ	0	$1-\gamma$	0	0	1
γ	0	$1-\gamma$	0	1	0

$$\begin{aligned}
\frac{\alpha}{\alpha\gamma-\alpha-\gamma}\lambda_3^1 + \frac{\gamma}{\alpha\gamma-\alpha-\gamma}(\lambda_2^2 + \lambda_3^2) \quad &\leq \quad -1 \\
-\frac{\alpha}{\alpha\beta-\alpha-\beta}\lambda_3^1 + \frac{\beta}{\alpha+\beta-\alpha\beta}(\lambda_2^2 + \lambda_3^2) \quad &\leq \quad 1
\end{aligned}$$

Case 7: $p_1^2 < p_1^1 < p_2^2, p_1^2 < p_2^1 < p_2^2, p_1^2 < p_3^1 < p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	α	0	$1-\alpha$
0	0	1	α	$1-\alpha$	0
0	1	0	β	0	$1-\beta$
0	1	0	β	$1-\beta$	0
1	0	0	γ	0	$1-\gamma$
1	0	0	γ	$1-\gamma$	0

$$\begin{aligned}
\frac{\alpha-\gamma}{\gamma-1}\lambda_3^1 + \frac{1}{\gamma-1}(\lambda_2^2 + \lambda_3^2) \quad &\leq \quad -1 \\
\frac{\beta-\alpha}{\beta-1}\lambda_3^1 + \frac{1}{1-\beta}(\lambda_2^2 + \lambda_3^2) \quad &\leq \quad 1
\end{aligned}$$

Case 8: $p_1^2 < p_1^1 < p_2^2, p_2^1 > p_2^2, p_1^2 < p_3^1 < p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
1	0	0	α	$1-\alpha$	0
1	0	0	α	0	$1-\alpha$
0	0	1	β	$1-\beta$	0
0	0	1	β	0	$1-\beta$
γ	$1-\gamma$	0	0	0	1
γ	$1-\gamma$	0	0	1	0
0	δ	$1-\delta$	0	0	1
0	δ	$1-\delta$	0	1	0

$$\begin{aligned}\frac{\beta-\alpha}{\alpha-1}\lambda_3^1+\frac{1}{\alpha-1}(\lambda_2^2+\lambda_3^2) &\leq -1\\ \frac{\beta}{\beta+\delta-\beta\delta}\lambda_3^1+\frac{\delta}{\beta+\delta-\beta\delta}(\lambda_2^2+\lambda_3^2) &\leq 1\end{aligned}$$

Case 9: $p_1^2 < p_1^1 < p_2^2, p_2^1 > p_2^2, p_3^1 > p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
1	0	0	α	$1-\alpha$	0
1	0	0	α	0	$1-\alpha$
β	$1-\beta$	0	0	0	1
β	$1-\beta$	0	0	1	0
γ	0	$1-\gamma$	0	0	1
γ	0	$1-\gamma$	0	1	0

$$\frac{\alpha}{(\alpha-1)(\gamma-1)}\lambda_3^1+\frac{1}{\alpha-1}(\lambda_2^2+\lambda_3^2) \quad\leq\quad -1$$

Case 10: $p_1^1 < p_1^2, p_1^2 < p_2^1 < p_2^2, p_1^2 < p_3^1 < p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	α	0	$1-\alpha$
0	0	1	α	$1-\alpha$	0
0	1	0	β	0	$1-\beta$
0	1	0	β	$1-\beta$	0
γ	0	$1-\gamma$	1	0	0
δ	$1-\delta$	0	1	0	0

$$\begin{aligned}\frac{1}{1-\gamma}\lambda_3^1-\frac{\gamma}{(1-\alpha)(1-\gamma)}(\lambda_2^2+\lambda_3^2) &\leq 1\\ \frac{\beta-\alpha}{\beta-1}\lambda_3^1+\frac{1}{1-\beta}(\lambda_2^2+\lambda_3^2) &\leq 1\end{aligned}$$

Case 11: $p_1^1 < p_1^2, p_2^1 > p_2^2, p_1^2 < p_3^1 < p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	α	0	$1-\alpha$
0	0	1	α	$1-\alpha$	0
0	β	$1-\beta$	0	0	1
0	β	$1-\beta$	0	1	0
γ	$1-\gamma$	0	0	0	1
γ	$1-\gamma$	0	0	1	0
δ	0	$1-\delta$	1	0	0
ϵ	$1-\epsilon$	0	1	0	0

$$\begin{aligned}\frac{1}{1-\delta}\lambda_3^1 - \frac{\delta}{(1-\alpha)(1-\delta)}(\lambda_2^2+\lambda_3^2) &\leq 1\\ \frac{\alpha}{\alpha+\beta-\alpha\beta}\lambda_3^1 + \frac{\beta}{\alpha+\beta-\alpha\beta}(\lambda_2^2+\lambda_3^2) &\leq 1\end{aligned}$$

Case 12: $p_1^1 < p_1^2, p_2^1 > p_2^2, p_3^1 > p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
α	$1-\alpha$	0	1	0	0
β	0	$1-\beta$	1	0	0
γ	$1-\gamma$	0	0	0	1
γ	$1-\gamma$	0	0	1	0
δ	0	$1-\delta$	0	0	1
δ	0	$1-\delta$	0	1	0

$$\frac{1}{1-\beta}\lambda_3^1 + \frac{\beta-\delta}{\beta-1}(\lambda_2^2+\lambda_3^2) \quad \leq 1$$

We now consider P_Δ in the situation which is described in Figure A.2.
General Assumption: $p_1^1 > p_2^1, p_2^1 < p_3^1, p_1^2 < p_2^2, p_2^2 = p_3^2$.
Case 1: $p_1^1 < p_1^2, p_1^2 < p_3^1 < p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	α	0	$1-\alpha$
0	0	1	α	$1-\alpha$	0
0	β	$1-\beta$	1	0	0
γ	0	$1-\gamma$	1	0	0

$$\begin{aligned}-\frac{1}{1-\gamma}\lambda_3^1 + \frac{\gamma}{(1-\gamma)(1-\alpha)}(\lambda_2^2+\lambda_3^2) &\leq -1\\ \frac{1}{1-\beta}\lambda_3^1 - \frac{\beta}{(1-\alpha)(1-\beta)}(\lambda_2^2+\lambda_3^2) &\leq 1\end{aligned}$$

Case 2: $p_1^1 < p_1^2, p_3^1 > p_2^2$

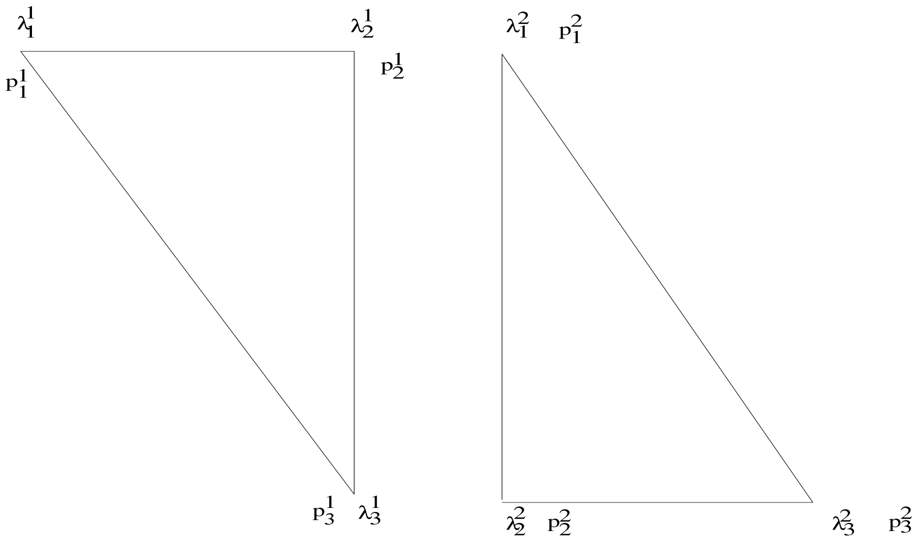

Figure A.2: Building vertices and facets of the polyhedron P_Δ

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	0	1
0	α	$1-\alpha$	0	1	0
0	β	$1-\beta$	1	0	0
γ	0	$1-\gamma$	0	0	1
γ	0	$1-\gamma$	0	1	0
δ	0	$1-\delta$	1	0	0

$$\begin{aligned}
\frac{1}{\delta-1}\lambda_3^1 + \frac{\gamma-\delta}{\delta-1}(\lambda_2^2+\lambda_3^2) &\leq -1\\
\frac{1}{1-\beta}\lambda_3^1 + \frac{\beta-\alpha}{\beta-1}(\lambda_2^2+\lambda_3^2) &\leq 1
\end{aligned}$$

Case 3: $p_1^2 < p_1^1 < p_2^2, p_2^1 < p_1^2, p_3^1 < p_1^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
α	0	$1-\alpha$	1	0	0
β	$1-\beta$	0	1	0	0
1	0	0	γ	0	$1-\gamma$
1	0	0	γ	$1-\gamma$	0

$$\frac{1}{1-\alpha}\lambda_3^1 + \frac{1}{1-\gamma}(\lambda_2^2+\lambda_3^2) \leq 1$$

Case 4: $p_1^2 < p_1^1 < p_2^2, p_2^1 < p_1^2, p_1^2 < p_3^1 < p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	α	0	$1-\alpha$
0	0	1	α	$1-\alpha$	0
0	β	$1-\beta$	1	0	0
γ	$1-\gamma$	0	1	0	0
1	0	0	δ	0	$1-\delta$
1	0	0	δ	$1-\delta$	0

$$\begin{aligned}\frac{1}{1-\beta}\lambda_3^1 - \frac{\beta}{(1-\alpha)(1-\beta)}(\lambda_2^2+\lambda_3^2) &\leq 1\\ \frac{\delta-\alpha}{\delta-1}\lambda_3^1 + \frac{1}{1-\delta}(\lambda_2^2+\lambda_3^2) &\leq 1\end{aligned}$$

Case 5: $p_1^2 < p_1^1 < p_2^2, p_2^1 < p_1^2, p_3^1 > p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	0	1
0	α	$1-\alpha$	0	1	0
0	β	$1-\beta$	1	0	0
γ	$1-\gamma$	0	1	0	0
δ	0	$1-\delta$	0	0	1
δ	0	$1-\delta$	0	1	0
1	0	0	ϵ	0	$1-\epsilon$
1	0	0	ϵ	$1-\epsilon$	0

$$\begin{aligned}-\frac{\epsilon}{(\delta-1)(\epsilon-1)}\lambda_3^1 + \frac{1}{1-\epsilon}(\lambda_2^2+\lambda_3^2) &\leq 1\\ \frac{1}{1-\beta}\lambda_3^1 + \frac{\beta-\alpha}{\beta-1}(\lambda_2^2+\lambda_3^2) &\leq 1\end{aligned}$$

Case 6: $p_1^2 < p_1^1 < p_2^2, p_1^2 < p_2^1 < p_2^2, p_1^2 < p_3^1 < p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	α	0	$1-\alpha$
0	0	1	α	$1-\alpha$	0
0	1	0	β	0	$1-\beta$
0	1	0	β	$1-\beta$	0
1	0	0	γ	0	$1-\gamma$
1	0	0	γ	$1-\gamma$	0

$$\begin{aligned}\frac{\alpha-\beta}{\beta-1}\lambda_3^1 + \frac{1}{\beta-1}(\lambda_2^2+\lambda_3^2) &\leq -1\\ \frac{\gamma-\alpha}{\gamma-1}\lambda_3^1 + \frac{1}{1-\gamma}(\lambda_2^2+\lambda_3^2) &\leq 1\end{aligned}$$

Case 7: $p_1^2 < p_1^1 < p_2^2, p_1^2 < p_2^1 < p_2^2, p_3^1 > p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	0	1
0	α	$1-\alpha$	0	1	0
0	1	0	β	0	$1-\beta$
0	1	0	β	$1-\beta$	0
γ	0	$1-\gamma$	0	0	1
γ	0	$1-\gamma$	0	1	0
1	0	0	δ	0	$1-\delta$
1	0	0	δ	$1-\delta$	0

$$\begin{aligned} \frac{\beta}{(\alpha-1)(\beta-1)}\lambda_3^1+\frac{1}{\beta-1}(\lambda_2^2+\lambda_3^2) &\leq -1 \\ -\frac{\delta}{(\gamma-1)(\delta-1)}\lambda_3^1+\frac{1}{1-\delta}(\lambda_2^2+\lambda_3^2) &\leq 1 \end{aligned}$$

Case 8: $p_1^1 > p_2^2, p_2^1 < p_1^2, p_3^1 < p_1^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
α	0	$1-\alpha$	1	0	0
β	$1-\beta$	0	1	0	0
γ	0	$1-\gamma$	0	0	1
γ	0	$1-\gamma$	0	1	0
δ	$1-\delta$	0	0	0	1
δ	$1-\delta$	0	0	1	0

$$\frac{1}{1-\alpha}\lambda_3^1+\frac{\alpha-\gamma}{\alpha-1}(\lambda_2^2+\lambda_3^2) \quad \leq 1$$

Case 9: $p_1^1 > p_2^2, p_2^1 < p_1^2, p_1^2 < p_3^1 < p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	α	0	$1-\alpha$
0	0	1	α	$1-\alpha$	0
0	β	$1-\beta$	1	0	0
γ	0	$1-\gamma$	0	0	1
γ	0	$1-\gamma$	0	1	0
δ	$1-\delta$	0	1	0	0
ϵ	$1-\epsilon$	0	0	0	1
ϵ	$1-\epsilon$	0	0	1	0

$$\begin{aligned} \frac{1}{1-\beta}\lambda_3^1-\frac{\beta}{(\alpha-1)(\beta-1)}(\lambda_2^2+\lambda_3^2) &\leq 1 \\ \frac{\alpha}{\alpha+\gamma-\alpha\gamma}\lambda_3^1+\frac{\gamma}{\alpha+\gamma-\alpha\gamma}(\lambda_2^2+\lambda_3^2) &\leq 1 \end{aligned}$$

Case 10: $p_1^1 > p_2^2, p_2^1 < p_1^2, p_3^1 > p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	0	1
0	α	$1-\alpha$	0	1	0
0	β	$1-\beta$	1	0	0
γ	$1-\gamma$	0	1	0	0
δ	$1-\delta$	0	0	0	1
δ	$1-\delta$	0	0	1	0

$$\frac{1}{1-\beta}\lambda_3^1 + \frac{\beta-\alpha}{\beta-1}(\lambda_2^2+\lambda_3^2) \quad \leq 1$$

Case 11: $p_1^1 > p_2^2, p_1^2 < p_2^1 < p_2^2, p_1^2 < p_3^1 < p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	α	0	$1-\alpha$
0	0	1	α	$1-\alpha$	0
0	1	0	β	0	$1-\beta$
0	1	0	β	$1-\beta$	0
γ	0	$1-\gamma$	0	0	1
γ	0	$1-\gamma$	0	1	0
δ	$1-\delta$	0	0	0	1
δ	$1-\delta$	0	0	1	0

$$\begin{aligned}
\frac{\alpha-\beta}{\beta-1}\lambda_3^1 + \frac{1}{\beta-1}(\lambda_2^2+\lambda_3^2) \quad &\leq \quad -1 \\
\frac{\alpha}{\alpha+\gamma-\alpha\gamma}\lambda_3^1 + \frac{\gamma}{\alpha+\gamma-\alpha\gamma}(\lambda_2^2+\lambda_3^2) \quad &\leq \quad 1
\end{aligned}$$

Case 12: $p_1^1 > p_2^2, p_1^2 < p_2^1 < p_2^2, p_3^1 > p_2^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	0	1
0	α	$1-\alpha$	0	1	0
0	1	0	β	0	$1-\beta$
0	1	0	β	$1-\beta$	0
γ	$1-\gamma$	0	0	0	1
γ	$1-\gamma$	0	0	1	0

$$\frac{\beta}{(\alpha-1)(\beta-1)}\lambda_3^1 + \frac{1}{\beta-1}(\lambda_2^2+\lambda_3^2) \quad \leq \quad -1$$

Let us consider now P_Δ in the situation which is described in Figure A.3.
General Assumption: $p_1^1 < p_2^1, p_2^1 > p_3^1, p_1^2 = p_2^2, p_2^2 < p_3^2$.
Case 1: $p_1^1 < p_1^2, p_1^2 < p_2^1 < p_3^2, p_3^1 < p_1^2$

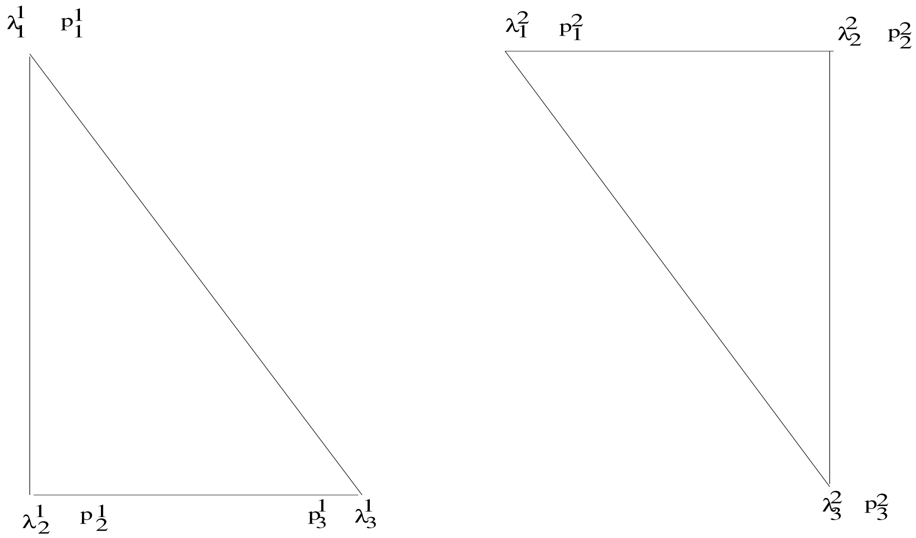

Figure A.3: Building vertices and facets of the polyhedron P_Δ

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	1	0
0	α	$1-\alpha$	1	0	0
0	1	0	0	β	$1-\beta$
0	1	0	β	0	$1-\beta$
γ	$1-\gamma$	0	0	1	0
γ	$1-\gamma$	0	1	0	0

$$\frac{1}{1-\alpha}\lambda_3^1 + \frac{1}{1-\beta}\lambda_3^2 \quad \leq 1$$

Case 2: $p_1^1 < p_1^2, p_1^2 < p_2^1 < p_3^2, p_1^2 < p_3^1 < p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	0	α	$1-\alpha$
0	0	1	α	0	$1-\alpha$
0	1	0	0	β	$1-\beta$
0	1	0	β	0	$1-\beta$
γ	0	$1-\gamma$	0	1	0
γ	0	$1-\gamma$	1	0	0
δ	$1-\delta$	0	0	1	0
δ	$1-\delta$	0	1	0	0

$$\begin{aligned} \frac{1}{1-\gamma}\lambda_3^1 - \frac{\gamma}{(\alpha-1)(\gamma-1)}\lambda_3^2 &\leq 1 \\ \frac{\beta-\alpha}{\beta-1}\lambda_3^1 + \frac{1}{1-\beta}\lambda_3^2 &\leq 1 \end{aligned}$$

Case 3: $p_1^1 < p_1^2, p_2^1 > p_3^2, p_3^1 < p_1^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	1	0
0	α	$1-\alpha$	1	0	0
0	β	$1-\beta$	0	0	1
γ	$1-\gamma$	0	0	0	1
δ	$1-\delta$	0	0	1	0
δ	$1-\delta$	0	1	0	0

$$\frac{1}{1-\alpha}\lambda_3^1 + \frac{\alpha-\beta}{\alpha-1}\lambda_3^2 \quad \leq 1$$

Case 4: $p_1^1 < p_1^2, p_2^1 > p_3^2, p_3^1 > p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
α	0	$1-\alpha$	0	0	1
β	$1-\beta$	0	0	0	1
γ	0	$1-\gamma$	0	1	0
γ	0	$1-\gamma$	1	0	0
δ	$1-\delta$	0	0	1	0
δ	$1-\delta$	0	1	0	0

$$\frac{1}{1-\gamma}\lambda_3^1 + \frac{\gamma-\alpha}{\gamma-1}\lambda_3^2 \quad \leq 1$$

Case 5: $p_1^1 < p_1^2, p_2^1 > p_3^2, p_1^2 < p_3^1 < p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	0	α	$1-\alpha$
0	0	1	α	0	$1-\alpha$
0	β	$1-\beta$	0	0	1
γ	0	$1-\gamma$	0	1	0
γ	0	$1-\gamma$	1	0	0
δ	$1-\delta$	0	0	0	1
ϵ	$1-\epsilon$	0	0	1	0
ϵ	$1-\epsilon$	0	1	0	0

$$\frac{1}{1-\gamma}\lambda_3^1 - \frac{\gamma}{(\alpha-1)(\gamma-1)}\lambda_3^2 \quad \leq 1$$
$$\frac{\alpha}{\alpha+\beta-\alpha\beta}\lambda_3^1 + \frac{\beta}{\alpha+\beta-\alpha\beta}\lambda_3^2 \quad \leq 1$$

Case 6: $p_1^2 < p_1^1 < p_3^2, p_1^2 < p_2^1 < p_3^2, p_3^1 < p_1^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	1	0
0	α	$1-\alpha$	1	0	0
0	1	0	0	β	$1-\beta$
0	1	0	β	0	$1-\beta$
γ	0	$1-\gamma$	0	1	0
γ	0	$1-\gamma$	1	0	0
1	0	0	0	δ	$1-\delta$
1	0	0	δ	0	$1-\delta$

$$\begin{aligned} \frac{1}{\gamma-1}\lambda_3^1 + \frac{1}{\delta-1}\lambda_3^2 &\leq -1 \\ \frac{1}{1-\alpha}\lambda_3^1 + \frac{1}{1-\beta}\lambda_3^2 &\leq 1 \end{aligned}$$

Case 7: $p_1^2 < p_1^1 < p_3^2, p_1^2 < p_2^1 < p_3^2, p_1^2 < p_3^1 < p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	0	α	$1-\alpha$
0	0	1	α	0	$1-\alpha$
0	1	0	0	β	$1-\beta$
0	1	0	β	0	$1-\beta$
1	0	0	0	γ	$1-\gamma$
1	0	0	γ	0	$1-\gamma$

$$\begin{aligned} \frac{\alpha-\gamma}{\gamma-1}\lambda_3^1 + \frac{1}{\gamma-1}\lambda_3^2 &\leq -1 \\ \frac{\beta-\alpha}{\beta-1}\lambda_3^1 + \frac{1}{1-\beta}\lambda_3^2 &\leq 1 \end{aligned}$$

Case 8: $p_1^2 < p_1^1 < p_3^2, p_2^1 > p_3^2, p_3^1 < p_1^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	1	0
0	α	$1-\alpha$	1	0	0
0	β	$1-\beta$	0	0	1
γ	0	$1-\gamma$	0	1	0
γ	0	$1-\gamma$	1	0	0
δ	$1-\delta$	0	0	0	1
1	0	0	0	ϵ	$1-\epsilon$
1	0	0	ϵ	0	$1-\epsilon$

$$\begin{aligned} \frac{1}{\gamma-1}\lambda_3^1 + \frac{1}{\epsilon-1}\lambda_3^2 &\leq -1 \\ \frac{1}{1-\alpha}\lambda_3^1 + \frac{\alpha-\beta}{\alpha-1}\lambda_3^2 &\leq 1 \end{aligned}$$

Case 9: $p_1^2 < p_1^1 < p_3^2, p_2^1 > p_3^2, p_1^2 < p_3^1 < p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	0	α	$1-\alpha$
0	0	1	α	0	$1-\alpha$
0	β	$1-\beta$	0	0	1
γ	$1-\gamma$	0	0	0	1
1	0	0	0	δ	$1-\delta$
1	0	0	δ	0	$1-\delta$

$$\begin{aligned} \frac{\alpha-\delta}{\delta-1}\lambda_3^1 + \frac{1}{\delta-1}\lambda_3^2 &\leq -1 \\ \frac{\alpha}{\alpha+\beta-\alpha\beta}\lambda_3^1 + \frac{\beta}{\alpha+\beta-\alpha\beta}\lambda_3^2 &\leq 1 \end{aligned}$$

Case 10: $p_1^2 < p_1^1 < p_3^2, p_2^1 > p_3^2, p_3^1 > p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
α	0	$1-\alpha$	0	0	1
β	$1-\beta$	0	0	0	1
1	0	0	0	γ	$1-\gamma$
1	0	0	γ	0	$1-\gamma$

$$\frac{\gamma}{(\alpha-1)(\gamma-1)}\lambda_3^1 - \frac{1}{1-\gamma}\lambda_3^2 \leq -1$$

Case 11: $p_1^1 > p_3^2, p_3^1 < p_1^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	1	0
0	α	$1-\alpha$	1	0	0
0	β	$1-\beta$	0	0	1
γ	0	$1-\gamma$	0	1	0
γ	0	$1-\gamma$	1	0	0
δ	0	$1-\delta$	0	0	1

$$\begin{aligned} \frac{1}{\alpha-1}\lambda_3^1 + \frac{\beta-\alpha}{\alpha-1}\lambda_3^2 &\leq -1 \\ \frac{1}{1-\gamma}\lambda_3^1 + \frac{\gamma-\delta}{\gamma-1}\lambda_3^2 &\leq 1 \end{aligned}$$

Case 12: $p_1^1 > p_3^2, p_1^2 < p_3^1 < p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	0	α	$1-\alpha$
0	0	1	α	0	$1-\alpha$
0	β	$1-\beta$	0	0	1
γ	0	$1-\gamma$	0	0	1

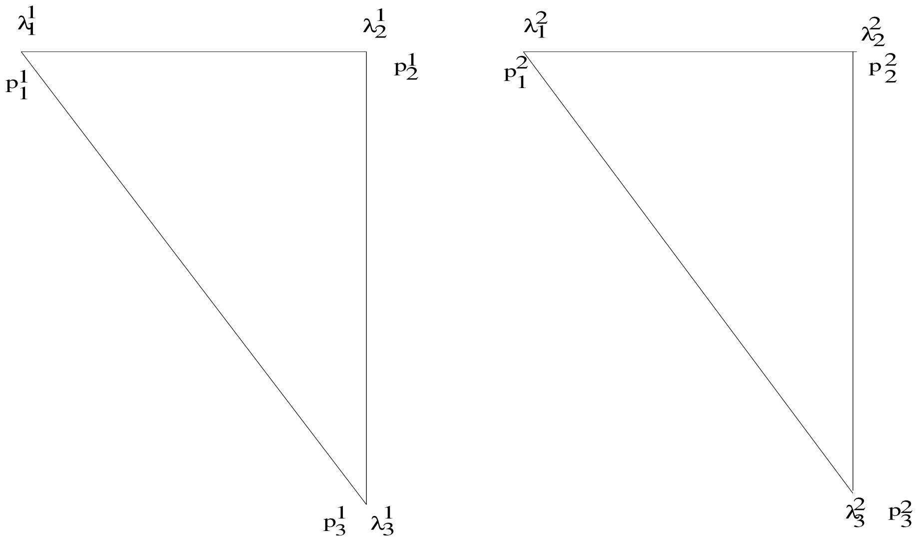

Figure A.4: Building vertices and facets of the polyhedron P_Δ

$$
\begin{aligned}
\frac{\alpha}{\alpha\gamma-\alpha-\gamma}\lambda_3^1+\frac{\gamma}{\alpha\gamma-\alpha-\gamma}\lambda_3^2 &\le -1\\
\frac{\alpha}{\alpha+\beta-\alpha\beta}\lambda_3^1+\frac{\beta}{\alpha+\beta-\alpha\beta}\lambda_3^2 &\le 1
\end{aligned}
$$

Finally we consider P_Δ in the situation which is described in Figure A.4.
General Assumption: $p_1^1 > p_2^1, p_2^1 < p_3^1, p_1^2 = p_2^2, p_2^2 < p_3^2$.
Case 1: $p_1^1 < p_1^2, p_1^2 < p_3^1 < p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	0	α	$1-\alpha$
0	0	1	α	0	$1-\alpha$
0	β	$1-\beta$	0	1	0
0	β	$1-\beta$	1	0	0
γ	0	$1-\gamma$	0	1	0
γ	0	$1-\gamma$	1	0	0

$$
\begin{aligned}
\frac{1}{\gamma-1}\lambda_3^1+\frac{\gamma}{(\alpha-1)(\gamma-1)}\lambda_3^2 &\le -1\\
\frac{1}{1-\beta}\lambda_3^1-\frac{\beta}{(\alpha-1)(\beta-1)}\lambda_3^2 &\le 1
\end{aligned}
$$

Case 2: $p_1^1 < p_1^2, p_3^1 > p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	0	1
0	β	$1-\beta$	0	1	0
0	β	$1-\beta$	1	0	0
γ	0	$1-\gamma$	0	0	1
δ	0	$1-\delta$	0	1	0
δ	0	$1-\delta$	1	0	0

$$\begin{aligned}\frac{1}{\delta-1}\lambda_3^1+\frac{\gamma-\delta}{\delta-1}\lambda_3^2 &\leq -1\\ \frac{1}{1-\beta}\lambda_3^1+\frac{\beta-\alpha}{\beta-1}\lambda_3^2 &\leq 1\end{aligned}$$

Case 3: $p_1^2 < p_1^1 < p_3^2, p_2^1 < p_1^2, p_3^1 < p_1^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
α	0	$1-\alpha$	0	1	0
α	0	$1-\alpha$	1	0	0
β	$1-\beta$	0	0	1	0
β	$1-\beta$	0	1	0	0
1	0	0	0	γ	$1-\gamma$
1	0	0	γ	0	$1-\gamma$

$$\frac{1}{1-\alpha}\lambda_3^1+\frac{1}{1-\gamma}\lambda_3^2 \leq 1$$

Case 4: $p_1^2 < p_1^1 < p_3^2, p_2^1 < p_1^2, p_1^2 < p_3^1 < p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	0	α	$1-\alpha$
0	0	1	α	0	$1-\alpha$
0	β	$1-\beta$	0	1	0
0	β	$1-\beta$	1	0	0
γ	$1-\gamma$	0	0	1	0
γ	$1-\gamma$	0	1	0	0
1	0	0	0	δ	$1-\delta$
1	0	0	δ	0	$1-\delta$

$$\begin{aligned}\frac{1}{1-\beta}\lambda_3^1-\frac{\beta}{(\alpha-1)(\beta-1)}\lambda_3^2 &\leq 1\\ \frac{\delta-\alpha}{\delta-1}\lambda_3^1+\frac{1}{1-\delta}\lambda_3^2 &\leq 1\end{aligned}$$

Case 5: $p_1^2 < p_1^1 < p_3^2, p_2^1 < p_1^2, p_3^1 > p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	0	1
0	β	$1-\beta$	0	1	0
0	β	$1-\beta$	1	0	0
γ	$1-\gamma$	0	0	1	0
γ	$1-\gamma$	0	1	0	0
δ	0	$1-\delta$	0	0	1
1	0	0	0	ϵ	$1-\epsilon$
1	0	0	ϵ	0	$1-\epsilon$

$$\begin{aligned}
-\frac{\epsilon}{(\delta-1)(\epsilon-1)}\lambda_3^1+\frac{1}{1-\epsilon}\lambda_3^2 &\leq 1\\
\frac{1}{1-\beta}\lambda_3^1+\frac{\beta-\alpha}{\beta-1}\lambda_3^2 &\leq 1
\end{aligned}$$

Case 6: $p_1^2 < p_1^1 < p_3^2, p_1^2 < p_2^1 < p_3^2, p_1^2 < p_3^1 < p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	0	α	$1-\alpha$
0	0	1	α	0	$1-\alpha$
0	1	0	0	β	$1-\beta$
0	1	0	β	0	$1-\beta$
1	0	0	0	γ	$1-\gamma$
1	0	0	γ	0	$1-\gamma$

$$\begin{aligned}
\frac{\alpha-\beta}{\beta-1}\lambda_3^1+\frac{1}{\beta-1}\lambda_3^2 &\leq -1\\
\frac{\gamma-\alpha}{\gamma-1}\lambda_3^1+\frac{1}{1-\gamma}\lambda_3^2 &\leq 1
\end{aligned}$$

Case 7: $p_1^2 < p_1^1 < p_3^2, p_1^2 < p_2^1 < p_3^2, p_3^1 > p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	0	1
0	1	0	0	β	$1-\beta$
0	1	0	β	0	$1-\beta$
γ	0	$1-\gamma$	0	0	1
1	0	0	0	δ	$1-\delta$
1	0	0	δ	0	$1-\delta$

$$\begin{aligned}
\frac{\beta}{(\alpha-1)(\beta-1)}\lambda_3^1+\frac{1}{\beta-1}\lambda_3^2 &\leq -1\\
-\frac{\delta}{(\gamma-1)(\delta-1)}\lambda_3^1+\frac{1}{1-\delta}\lambda_3^2 &\leq 1
\end{aligned}$$

Case 8: $p_1^1 > p_3^2, p_2^1 < p_1^2, p_3^1 < p_1^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
α	0	$1-\alpha$	0	1	0
α	0	$1-\alpha$	1	0	0
β	$1-\beta$	0	0	1	0
β	$1-\beta$	0	1	0	0
γ	0	$1-\gamma$	0	0	1
δ	$1-\delta$	0	0	0	1

$$\frac{1}{1-\alpha}\lambda_3^1 + \frac{\alpha-\gamma}{\alpha-1}\lambda_3^2 \quad \leq 1$$

Case 9: $p_1^1 > p_3^2, p_2^1 < p_1^2, p_1^2 < p_3^1 < p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	0	α	$1-\alpha$
0	0	1	α	0	$1-\alpha$
0	β	$1-\beta$	0	1	0
0	β	$1-\beta$	1	0	0
γ	0	$1-\gamma$	0	0	1
δ	$1-\delta$	0	0	1	0
δ	$1-\delta$	0	1	0	0
ϵ	$1-\epsilon$	0	0	0	1

$$\begin{aligned} \frac{1}{1-\beta}\lambda_3^1 - \frac{\beta}{(\alpha-1)(\beta-1)}\lambda_3^2 &\leq 1 \\ \frac{\alpha}{\alpha+\gamma-\alpha\gamma}\lambda_3^1 + \frac{\gamma}{\alpha+\gamma-\alpha\gamma}\lambda_3^2 &\leq 1 \end{aligned}$$

Case 10: $p_1^1 > p_3^2, p_2^1 < p_1^2, p_3^1 > p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	0	1
0	β	$1-\beta$	0	1	0
0	β	$1-\beta$	1	0	0
γ	$1-\gamma$	0	0	1	0
γ	$1-\gamma$	0	1	0	0
δ	$1-\delta$	0	0	0	1

$$\frac{1}{1-\beta}\lambda_3^1 + \frac{\beta-\alpha}{\beta-1}\lambda_3^2 \quad \leq 1$$

Case 11: $p_1^1 > p_3^2, p_1^2 < p_2^1 < p_3^2, p_1^2 < p_3^1 < p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	0	1	0	α	$1-\alpha$
0	0	1	α	0	$1-\alpha$
0	1	0	0	β	$1-\beta$
0	1	0	β	0	$1-\beta$
γ	0	$1-\gamma$	0	0	1
δ	$1-\delta$	0	0	0	1

$$\begin{aligned} \frac{\alpha-\beta}{\beta-1}\lambda_3^1 + \frac{1}{\beta-1}\lambda_3^2 &\leq -1 \\ \frac{\alpha}{\alpha+\gamma-\alpha\gamma}\lambda_3^1 + \frac{\gamma}{\alpha+\gamma-\alpha\gamma}\lambda_3^2 &\leq 1 \end{aligned}$$

Case 12: $p_1^1 > p_3^2, p_1^2 < p_2^1 < p_3^2, p_3^1 > p_3^2$

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
0	α	$1-\alpha$	0	0	1
0	1	0	0	β	$1-\beta$
0	1	0	β	0	$1-\beta$
γ	$1-\gamma$	0	0	0	1

$$\frac{\beta}{(\alpha-1)(\beta-1)}\lambda_3^1 + \frac{1}{\beta-1}\lambda_3^2 \quad \leq -1$$

We remark that it is unfortunately not easy to simplify this classification of the facets. Here we consider a simple example: For Figure A.1, Case 3 the vertices

Ingoing pipe			Outgoing pipe		
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2
γ	0	$1-\gamma$	0	0	1
γ	0	$1-\gamma$	0	1	0
δ	0	$1-\delta$	1	0	0

lead to the facet

$$\frac{1}{\delta-1}\lambda_3^1 + \frac{\gamma-\delta}{\delta-1}(\lambda_2^2+\lambda_3^2) \quad \leq -1.$$

In Figure A.2, Case 8 the same vertices lead to the facet

$$\frac{1}{\delta-1}\lambda_3^1 + \frac{\gamma-\delta}{\delta-1}(\lambda_2^2+\lambda_3^2) \quad \geq -1.$$

We see that we need **all** vertices in the particular situations in order to define the vertices and not only the vertices that fulfil the facets at equality.
Let us give a little

Lemma 45 *The inequalities we have given in Cases* $1, 2, \ldots, 12$ *for each of the four situations (which have been defined by Figures* $1, 2, 3, 4$*) are indeed facets of* $P_\triangle$ *in the special situation.*

Proof. The principle idea of the proofs is very simple:
We have already seen that in order to show the validity of the inequalities it is sufficient to show that the vertices fulfil the inequality.
Moreover we know that the dimension of the considered polyhedra can be maximal 3 which is easy to see. The polyhedra are a subspace of $\mathbb{R}^6$ and every point of each polyhedron is defined by 3 equalities. So the maximal dimension of the polyhedra can be $6-3=3$. Here it is important to know that in the situations we discussed the three equalities that are defining the polyhedron $P_\triangle$ are linearly independent. For every polyhedron we can show that at least 3 vertices fulfil the inequality at equality. Now it is clear that we have found a facet of the polyhedron.

As an example we prove Case 8 and Case 11 for the triangle combination defined by Figure 1, the other proofs are in a analogous manner:

Remember that we get in Case 8 the following vertices:

Ingoing pipe			Outgoing pipe			Nr.
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2	
1	0	0	α	$1-\alpha$	0	1
1	0	0	α	0	$1-\alpha$	2
0	0	1	β	$1-\beta$	0	3
0	0	1	β	0	$1-\beta$	4
γ	$1-\gamma$	0	0	0	1	5
γ	$1-\gamma$	0	0	1	0	6
0	δ	$1-\delta$	0	0	1	7
0	δ	$1-\delta$	0	1	0	8

First consider the inequality:

$$\frac{\beta-\alpha}{\alpha-1}\lambda_3^1 + \frac{1}{\alpha-1}(\lambda_2^2+\lambda_3^2) \quad \leq \quad -1.$$

Calculation for point 1 and 2:

$$0+\frac{1}{\alpha-1}(1-\alpha) = -1.$$

Calculation for point 3 and 4:

$$\frac{\beta-\alpha}{\alpha-1}+\frac{1-\beta}{\alpha-1} = \frac{1-\alpha}{\alpha-1} = -1.$$

Calculation for point 5 and 6:

$$0+\frac{1}{\alpha-1} = -\frac{1}{1-\alpha} < -1,$$

since $0<\alpha<1$.
Calculation for point 7 and 8:

$$\frac{\beta-\alpha}{\alpha-1}(1-\delta)+\frac{1}{\alpha-1} = \frac{\beta-\beta\delta-\alpha+\alpha\delta+1}{\alpha-1} = -1+\frac{\beta-\beta\delta+\alpha\delta}{\alpha-1} = -1-\frac{\beta(1-\delta)+\alpha\delta}{1-\alpha} < -1.$$

For the last estimate remember $\beta>0, \delta>0, \alpha<1$.
Let us consider now the second inequality:

$$\frac{\beta}{\beta+\delta-\beta\delta}\lambda_3^1 + \frac{\delta}{\beta+\delta-\beta\delta}(\lambda_2^2+\lambda_3^2) \quad \leq \quad 1.$$

We remark that $0<\beta<1$ and $0<\delta<1$ implies $\beta+\delta-\beta\delta>0$.
Calculation for point 1 and 2:

$$0<\alpha\delta+\beta(1-\delta) \;\Rightarrow \delta-\alpha\delta<\beta+\delta-\delta\beta \;\Rightarrow \frac{\delta}{\beta+\delta-\beta\delta}(1-\alpha)<1.$$

Calculation for point 3 and 4:

$$\frac{\beta}{\beta+\delta-\beta\delta}+\frac{\delta}{\beta+\delta-\beta\delta}(1-\beta) = \frac{\beta+\delta(1-\beta)}{\beta+\delta-\beta\delta} = 1.$$

Calculation for point 5 and 6:

$$0+\frac{\delta}{\beta+\delta-\beta\delta} < \frac{\delta}{\beta\delta+\delta-\beta\delta} = \frac{\delta}{\delta} = 1.$$

This is clear because of $0 < \delta < 1$ and $0 < \beta < 1$.
Calculation for point 7 and 8:

$$\frac{\beta}{\beta+\delta-\beta\delta}(1-\delta)+\frac{\delta}{\beta+\delta-\beta\delta}=1.$$

We have shown that the two inequalities are indeed facets!

As a second example let us sketch the proof in Case 11:

Ingoing pipe			Outgoing pipe			Nr.
λ_1^1	λ_2^1	λ_3^1	λ_1^2	λ_2^2	λ_3^2	
0	0	1	α	0	$1-\alpha$	1
0	0	1	α	$1-\alpha$	0	2
0	β	$1-\beta$	0	0	1	3
0	β	$1-\beta$	0	1	0	4
γ	$1-\gamma$	0	0	0	1	5
γ	$1-\gamma$	0	0	1	0	6
δ	0	$1-\delta$	1	0	0	7
ϵ	$1-\epsilon$	0	1	0	0	8

Let us consider the first inequality:

$$\frac{1}{1-\delta}\lambda_3^1-\frac{\delta}{(1-\alpha)(1-\delta)}(\lambda_2^2+\lambda_3^2)\quad\leq 1.$$

The calculations are quite simple. Only the calculation for the point 3 and 4 is interesting:

$$\frac{1-\beta}{1-\delta}-\frac{\delta}{(1-\alpha)(1-\delta)}=\frac{(1-\beta)(1-\alpha)-\delta}{(1-\alpha)(1-\delta)}.$$

We know

$$\frac{\delta}{\beta}>0$$

and from $0<\alpha<1$ concludes $\frac{\alpha-1}{\alpha}<0$ and so it holds

$$\frac{\delta}{\beta}>\frac{\alpha-1}{\alpha}.$$

Now we can easily conclude

$$\delta\alpha>\beta(\alpha-1)\Rightarrow\ 1-\alpha-\delta+\alpha\delta>1-\alpha-\delta+\beta\alpha-\beta\Rightarrow\ (1-\alpha)(1-\delta)>(1-\beta)(1-\alpha)-\delta.$$

In summary we get

$$\frac{(1-\beta)(1-\alpha)-\delta}{(1-\alpha)(1-\delta)}<1.$$

Let us consider the second inequality:

$$\frac{\alpha}{\alpha+\beta-\alpha\beta}\lambda_3^1+\frac{\beta}{\alpha+\beta-\alpha\beta}(\lambda_2^2+\lambda_3^2)\quad\leq 1.$$

Only the calculation for point 7 is new and nontrivial:

$$\frac{\alpha}{\alpha+\beta-\alpha\beta}(1-\delta)=\frac{\alpha(1-\delta)}{\alpha(1-\beta)+\beta}<\frac{\alpha(1-\delta)}{\alpha(1-\beta)+\alpha\beta}=1-\delta<1.$$

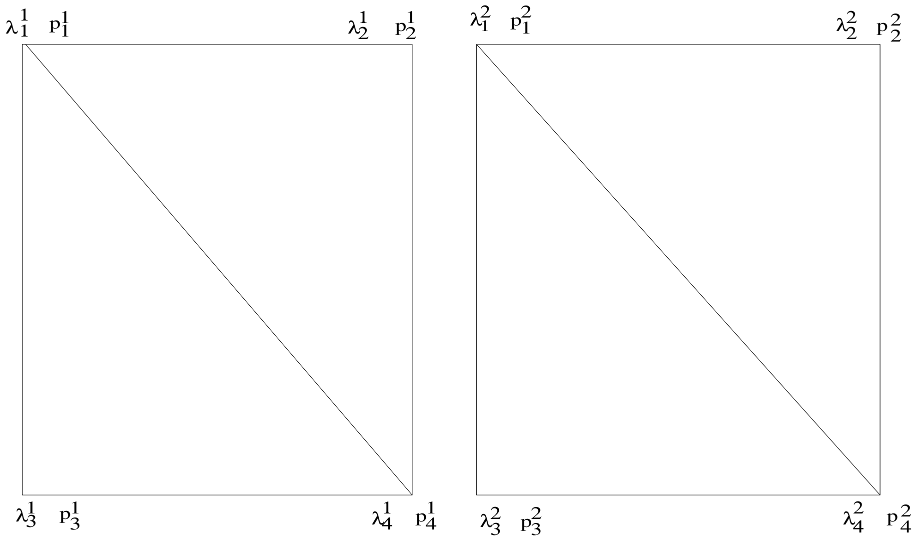

Figure A.5: Building vertices and facets of the polyhedron P_Δ

In this way all proofs in all other cases can be managed and we do not need any new idea for the proof of the other facets. □

Small tests showed that it is not sufficient to know the facets in the cases we have already examined. So we look at some complexer cases which are shown in Figure A.5. We only have to remark that now holds

$$\lambda^1 = \begin{pmatrix} \lambda_1^1 \\ \lambda_2^1 \\ \lambda_3^1 \\ \lambda_4^1 \end{pmatrix}, \lambda^2 = \begin{pmatrix} \lambda_1^2 \\ \lambda_2^2 \\ \lambda_3^2 \\ \lambda_4^2 \end{pmatrix}, p_{out}^1 = \begin{pmatrix} p_{out,1}^1 \\ p_{out,2}^1 \\ p_{out,3}^1 \\ p_{out,4}^1 \end{pmatrix}, p_{in}^2 = \begin{pmatrix} p_{in,1}^2 \\ p_{in,2}^2 \\ p_{in,3}^2 \\ p_{in,4}^2 \end{pmatrix}$$

where $p_{out}^1, p_{in}^2 \in \mathbb{R}^4$ are the two vectors of pressure values at the grid points described by the λ-variables λ^1, λ^2. Since the description of the problem even in this case is quite complex we restricted us on the calculation of valid inequalities. In the several cases it is easy but longwinded to control whether the calculated valid inequalities are indeed facets.

General Assumption: $p_1^1 > p_2^1, p_3^1 > p_4^1, p_1^1 < p_3^1, p_2^1 < p_4^2, p_1^2 = p_2^2, p_3^2 = p_4^2, p_1^2 < p_3^2, p_2^2 < p_4^2$.
Case 1: $p_1^1 < p_1^2, p_1^2 < p_3^1 < p_3^2, p_4^1 < p_1^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	0	α	$1-\alpha$	0	1	0	0
0	0	α	$1-\alpha$	1	0	0	0
0	0	1	0	0	β	0	$1-\beta$
0	0	1	0	β	0	0	$1-\beta$
0	0	1	0	β	0	$1-\beta$	0
γ	0	$1-\gamma$	0	0	1	0	0
γ	0	$1-\gamma$	0	1	0	0	0

$$\lambda_2^1 \leq 0$$
$$\lambda_2^2 + \frac{1}{1-\beta}\lambda_3^2 + \lambda_4^2 \leq 1$$
$$\frac{1}{1-\alpha}\lambda_4^1 + \frac{1}{1-\beta}(\lambda_3^2 + \lambda_4^2) \leq 1$$

Case 2: $p_1^1 < p_1^2, p_1^2 < p_3^1 < p_3^2, p_1^2 < p_4^1 < p_3^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	0	0	1	0	α	0	$1-\alpha$
0	0	0	1	α	0	0	$1-\alpha$
0	0	0	1	α	0	$1-\alpha$	0
0	0	1	0	0	β	0	$1-\beta$
0	0	1	0	β	0	0	$1-\beta$
0	0	1	0	β	0	$1-\beta$	0
0	γ	0	$1-\gamma$	0	1	0	0
0	γ	0	$1-\gamma$	1	0	0	0
δ	0	0	$1-\delta$	0	1	0	0
δ	0	0	$1-\delta$	1	0	0	0
ϵ	0	$1-\epsilon$	0	0	1	0	0
ϵ	0	$1-\epsilon$	0	1	0	0	0

$$\lambda_2^2 + \frac{1}{1-\beta}\lambda_3^2 + \lambda_4^2 \leq 1$$
$$\frac{\alpha-\beta}{\alpha-1}\lambda_3^1 + \lambda_2^2 + \frac{1}{1-\alpha}\lambda_3^2 + \lambda_4^2 \leq 1$$
$$\frac{1}{1-\epsilon}\lambda_3^1 + \frac{1}{1-\gamma}\lambda_4^1 - \frac{\epsilon}{(\beta-1)(\epsilon-1)}(\lambda_3^2 + \lambda_4^2) \leq 1$$
$$\frac{1}{\epsilon-1}\lambda_3^1 + \frac{1}{\delta-1}\lambda_4^1 + \frac{\epsilon}{(\beta-1)(\epsilon-1)}(\lambda_3^2 + \lambda_4^2) \leq -1$$
$$(1 + \frac{(1-\beta)\gamma}{(\alpha-1)(\gamma-1)})\lambda_3^1 + \frac{1}{1-\gamma}\lambda_4^1 - \frac{\gamma}{(\alpha-1)(\gamma-1)}(\lambda_3^2 + \lambda_4^2) \leq 1$$

Case 3: $p_1^1 < p_1^2, p_3^1 > p_3^2, p_4^1 < p_1^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	0	α	$1-\alpha$	0	1	0	0
0	0	α	$1-\alpha$	1	0	0	0
0	0	β	$1-\beta$	0	0	0	1
0	0	β	$1-\beta$	0	0	1	0
γ	0	$1-\gamma$	0	0	0	0	1
γ	0	$1-\gamma$	0	0	0	1	0
δ	0	$1-\delta$	0	0	1	0	0
δ	0	$1-\delta$	0	1	0	0	0

$$\lambda_2^1 \leq 0$$
$$\frac{1}{1-\alpha}\lambda_4^1 + \frac{\alpha-\beta}{\alpha-1}(\lambda_3^2 + \lambda_4^2) \leq 1$$

Case 4: $p_1^1 < p_1^2, p_3^1 > p_3^2, p_1^2 < p_4^1 < p_3^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	0	0	1	0	α	0	$1-\alpha$
0	0	0	1	α	0	0	$1-\alpha$
0	0	0	1	α	0	$1-\alpha$	0
0	0	β	$1-\beta$	0	0	0	1
0	0	β	$1-\beta$	0	0	1	0
0	γ	0	$1-\gamma$	0	1	0	0
0	γ	0	$1-\gamma$	1	0	0	0
δ	0	0	$1-\delta$	0	1	0	0
δ	0	0	$1-\delta$	1	0	0	0
ϵ	0	$1-\epsilon$	0	0	0	0	1
ϵ	0	$1-\epsilon$	0	0	0	1	0
ζ	0	$1-\zeta$	0	0	1	0	0
ζ	0	$1-\zeta$	0	1	0	0	0

$$\begin{aligned}
\frac{\alpha}{\beta(\alpha-1)}\lambda_3^1 + \lambda_2^2 + \frac{1}{1-\alpha}\lambda_3^2 + \lambda_4^2 &\leq 1\\
\frac{1}{1-\zeta}\lambda_3^1 + \frac{1}{1-\gamma}\lambda_4^1 + \frac{\zeta-\epsilon}{\zeta-1}(\lambda_3^2+\lambda_4^2) &\leq 1\\
\text{if } \alpha(\zeta-\epsilon)+\epsilon-\zeta(1+\gamma)\leq 0 \text{ and } \gamma(1-\beta-\epsilon)+\epsilon+\zeta(\beta-1)\leq 0 &\\
\frac{\beta-\alpha\beta+\alpha\gamma}{(\alpha-1)\beta(\gamma-1)}\lambda_3^1 + \frac{1}{1-\gamma}\lambda_4^1 - \frac{\gamma}{(\alpha-1)(\gamma-1)}(\lambda_3^2+\lambda_4^2) &\leq 1
\end{aligned}$$

Case 5: $p_1^1 < p_1^2, p_3^1 > p_3^2, p_4^1 > p_3^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	α	0	$1-\alpha$	0	0	0	1
0	α	0	$1-\alpha$	0	0	1	0
0	β	0	$1-\beta$	0	1	0	0
0	β	0	$1-\beta$	1	0	0	0
γ	0	0	$1-\gamma$	0	0	0	1
γ	0	0	$1-\gamma$	0	0	1	0
δ	0	$1-\delta$	0	0	0	0	1
δ	0	$1-\delta$	0	0	0	1	0
ϵ	0	0	$1-\epsilon$	0	1	0	0
ϵ	0	0	$1-\epsilon$	1	0	0	0
ζ	0	$1-\zeta$	0	0	1	0	0
ζ	0	$1-\zeta$	0	1	0	0	0

$$\begin{aligned}
\frac{1}{1-\zeta}\lambda_3^1 + \frac{1}{1-\beta}\lambda_4^1 + \frac{\zeta-\delta}{\zeta-1}(\lambda_3^2+\lambda_4^2) &\leq 1 \text{ if } \zeta(1-\alpha)\leq\beta-\alpha+\delta(1-\beta)\\
\frac{1-\alpha}{(\beta-1)(\delta-1)}\lambda_3^1 + \frac{1}{1-\beta}\lambda_4^1 + \frac{\beta-\alpha}{\beta-1}(\lambda_3^2+\lambda_4^2) &\leq 1 \text{ if } \frac{1-\alpha}{1-\beta}\leq\frac{1-\delta}{1-\zeta}\\
\frac{\gamma-1}{(\delta-1)(\epsilon-1)}\lambda_3^1 + \frac{1}{\epsilon-1}\lambda_4^1 + \frac{\gamma-\epsilon}{\epsilon-1}(\lambda_3^2+\lambda_4^2) &\leq -1
\end{aligned}$$

Case 6: $p_1^2 < p_1^1 < p_3^2, p_2^1 < p_1^2, p_1^2 < p_3^1 < p_3^2, p_4^1 < p_1^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	0	α	$1-\alpha$	0	1	0	0
0	0	α	$1-\alpha$	1	0	0	0
0	0	1	0	0	β	0	$1-\beta$
0	0	1	0	β	0	0	$1-\beta$
0	0	1	0	β	0	$1-\beta$	0
γ	0	0	$1-\gamma$	0	1	0	0
γ	0	0	$1-\gamma$	1	0	0	0
δ	$1-\delta$	0	0	0	1	0	0
δ	$1-\delta$	0	0	1	0	0	0
1	0	0	0	0	ϵ	0	$1-\epsilon$
1	0	0	0	ϵ	0	0	$1-\epsilon$
1	0	0	0	ϵ	0	$1-\epsilon$	0

$$\begin{aligned}
\frac{1-\alpha}{\alpha}\lambda_3^1 - \lambda_4^1 - \frac{1-\alpha}{\alpha(1-\beta)}(\lambda_3^2+\lambda_4^2) &\leq 0\\
\lambda_2^2 + \frac{1}{1-\beta}\lambda_3^2 + \lambda_4^2 &\leq 1\\
\frac{\epsilon-\beta}{\epsilon-1}\lambda_3^1 + \lambda_2^2 + \frac{1}{1-\epsilon}\lambda_3^2 + \lambda_4^2 &\leq 1\\
\frac{\gamma-\alpha}{\alpha(\gamma-1)}\lambda_3^1 + \frac{1}{1-\gamma}\lambda_4^1 + \frac{\gamma-\alpha\gamma}{\alpha(\beta-1)(\gamma-1)}(\lambda_3^2+\lambda_4^2) &\leq 1
\end{aligned}$$

Case 7: $p_1^2 < p_1^1 < p_3^2, p_2^1 < p_1^2, p_1^2 < p_3^1 < p_3^2, p_1^2 < p_4^1 < p_3^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	0	0	1	0	α	0	$1-\alpha$
0	0	0	1	α	0	0	$1-\alpha$
0	0	0	1	α	0	$1-\alpha$	0
0	0	1	0	0	β	0	$1-\beta$
0	0	1	0	β	0	0	$1-\beta$
0	0	1	0	β	0	$1-\beta$	0
0	γ	0	$1-\gamma$	0	1	0	0
0	γ	0	$1-\gamma$	1	0	0	0
δ	$1-\delta$	0	0	0	1	0	0
δ	$1-\delta$	0	0	1	0	0	0
1	0	0	0	0	ϵ	0	$1-\epsilon$
1	0	0	0	ϵ	0	0	$1-\epsilon$
1	0	0	0	ϵ	0	$1-\epsilon$	0

$$
\begin{aligned}
\lambda_3^1 - \frac{1}{1-\beta}(\lambda_3^2+\lambda_4^2) &\leq 0 \\
\lambda_2^2 + \frac{1}{1-\beta}\lambda_3^2 + \lambda_4^2 &\leq 1 \\
\frac{\beta-\epsilon}{1-\epsilon}\lambda_3^1 + \lambda_2^2 + \frac{1}{1-\epsilon}\lambda_3^2 + \lambda_4^2 &\leq 1 \text{ if } \alpha \geq \epsilon \\
(1+\frac{(1-\beta)\gamma}{(\alpha-1)(\gamma-1)})\lambda_3^1 + \frac{1}{1-\gamma}\lambda_4^1 - \frac{\gamma}{(\alpha-1)(\gamma-1)}(\lambda_3^2+\lambda_4^2) &\leq 1 \\
\frac{\epsilon-\beta}{\epsilon-1}\lambda_3^1 + \frac{\epsilon-\alpha}{\epsilon-1}\lambda_4^1 + \frac{1}{1-\epsilon}(\lambda_3^2+\lambda_4^2) &\leq 1
\end{aligned}
$$

Case 8: $p_1^2 < p_1^1 < p_3^2, p_2^1 < p_1^2, p_3^1 > p_3^2, p_4^1 < p_1^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	0	α	$1-\alpha$	0	1	0	0
0	0	α	$1-\alpha$	1	0	0	0
0	0	β	$1-\beta$	0	0	0	1
0	0	β	$1-\beta$	0	0	1	0
γ	0	0	$1-\gamma$	0	1	0	0
γ	0	0	$1-\gamma$	1	0	0	0
δ	0	$1-\delta$	0	0	0	0	1
δ	0	$1-\delta$	0	0	0	1	0
ϵ	$1-\epsilon$	0	0	0	1	0	0
ϵ	$1-\epsilon$	0	0	1	0	0	0
1	0	0	0	0	ζ	0	$1-\zeta$
1	0	0	0	ζ	0	0	$1-\zeta$
1	0	0	0	ζ	0	$1-\zeta$	0

$$
\begin{aligned}
\frac{1-\alpha}{\alpha}\lambda_3^1 - \lambda_4^1 + \frac{\alpha-\beta}{\alpha}(\lambda_3^2+\lambda_4^2) &\leq 0 \text{ if } \frac{1-\beta}{1-\alpha} \leq \delta \\
\frac{1}{1-\delta}\lambda_3^1 - \frac{\beta+\delta-1}{(1-\delta)(1-\beta)}\lambda_4^1 - (\lambda_3^2+\lambda_4^2) &\leq 0 \text{ if } \frac{1-\beta}{1-\alpha} \leq \delta \\
-\frac{\zeta}{(\delta-1)(\zeta-1)}\lambda_3^1 + \lambda_2^2 + \frac{1}{1-\zeta}\lambda_3^2 + \lambda_4^2 &\leq 1
\end{aligned}
$$

Case 9: $p_1^2 < p_1^1 < p_3^2, p_2^1 < p_1^2, p_3^1 > p_3^2, p_1^2 < p_4^1 < p_3^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	0	0	1	0	α	0	$1-\alpha$
0	0	0	1	α	0	0	$1-\alpha$
0	0	0	1	α	0	$1-\alpha$	0
0	0	β	$1-\beta$	0	0	0	1
0	0	β	$1-\beta$	0	0	1	0
0	γ	0	$1-\gamma$	0	1	0	0
0	γ	0	$1-\gamma$	1	0	0	0
δ	0	$1-\delta$	0	0	0	0	1
δ	0	$1-\delta$	0	0	0	1	0
ϵ	$1-\epsilon$	0	0	0	1	0	0
ϵ	$1-\epsilon$	0	0	1	0	0	0
1	0	0	0	0	ζ	0	$1-\zeta$
1	0	0	0	ζ	0	0	$1-\zeta$
1	0	0	0	ζ	0	$1-\zeta$	0

$$
\begin{aligned}
\frac{1}{\beta}\lambda_3^1 - (\lambda_3^2 + \lambda_4^2) &\leq 0 \text{ if } 1-\delta \leq \beta \\
\frac{1}{1-\delta}\lambda_3^1 + \frac{1-\beta-\delta}{(1-\delta)(1-\beta)}\lambda_4^1 - (\lambda_3^2 + \lambda_4^2) &\leq 0 \text{ if } 1-\delta \leq \beta \\
-\frac{\zeta}{(\zeta-1)(\delta-1)}\lambda_3^1 + \lambda_2^2 + \frac{1}{1-\zeta}\lambda_3^2 + \lambda_4^2 &\leq 1 \text{ if } \alpha \geq \zeta \text{ and } \zeta\beta \geq \delta(1-\zeta) \\
\frac{\beta - \alpha\beta + \alpha\gamma}{(\alpha-1)\beta(\gamma-1)}\lambda_3^1 + \frac{1}{1-\gamma}\lambda_4^1 - \frac{\gamma}{(\alpha-1)(\gamma-1)}(\lambda_3^2 + \lambda_4^2) &\leq 1
\end{aligned}
$$

Case 10: $p_1^2 < p_1^1 < p_3^2, p_2^1 < p_1^2, p_3^1 > p_3^2, p_4^1 > p_3^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	α	0	$1-\alpha$	0	0	0	1
0	α	0	$1-\alpha$	0	0	1	0
0	β	0	$1-\beta$	0	1	0	0
0	β	0	$1-\beta$	1	0	0	0
γ	0	0	$1-\gamma$	0	0	0	1
γ	0	0	$1-\gamma$	0	0	1	0
δ	0	$1-\delta$	0	0	0	0	1
δ	0	$1-\delta$	0	0	0	1	0
ϵ	$1-\epsilon$	0	0	0	1	0	0
ϵ	$1-\epsilon$	0	0	1	0	0	0
1	0	0	0	0	ζ	0	$1-\zeta$
1	0	0	0	ζ	0	0	$1-\zeta$
1	0	0	0	ζ	0	$1-\zeta$	0

$$\begin{aligned}
\frac{1}{1-\delta}\lambda_3^1 - (\lambda_3^2 + \lambda_4^2) &\leq 0 \\
-\frac{\zeta}{(\delta-1)(\zeta-1)}\lambda_3^1 - \frac{\zeta}{(\gamma-1)(\zeta-1)}\lambda_4^1 + \lambda_2^2 + \frac{1}{1-\zeta}\lambda_3^2 + \lambda_4^2 &\leq 1 \\
\frac{1-\alpha}{(\beta-1)(\delta-1)}\lambda_3^1 + \frac{1}{1-\beta}\lambda_4^1 + \frac{\beta-\alpha}{\beta-1}(\lambda_3^2 + \lambda_4^2) &\leq 1 \\
-\frac{\zeta}{(\delta-1)(\zeta-1)}\lambda_3^1 - \frac{\zeta}{(\gamma-1)(\zeta-1)}\lambda_4^1 + \frac{1}{1-\zeta}(\lambda_3^2 + \lambda_4^2) &\leq 1
\end{aligned}$$

Case 11: $p_1^2 < p_1^1 < p_3^2, p_1^2 < p_2^1 < p_3^2, p_1^2 < p_3^1 < p_3^2, p_1^2 < p_4^1 < p_3^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	0	0	1	0	α	0	$1-\alpha$
0	0	0	1	α	0	0	$1-\alpha$
0	0	0	1	α	0	$1-\alpha$	0
0	0	1	0	0	β	0	$1-\beta$
0	0	1	0	β	0	0	$1-\beta$
0	0	1	0	β	0	$1-\beta$	0
0	1	0	0	0	γ	0	$1-\gamma$
0	1	0	0	γ	0	0	$1-\gamma$
0	1	0	0	γ	0	$1-\gamma$	0
1	0	0	0	0	δ	0	$1-\delta$
1	0	0	0	δ	0	0	$1-\delta$
1	0	0	0	δ	0	$1-\delta$	0

$$\begin{aligned}
\lambda_2^2 - \frac{\gamma}{1-\gamma}\lambda_4^2 &\leq 0 \\
\lambda_2^2 + \frac{1}{1-\beta}\lambda_3^2 + \lambda_4^2 &\leq 1 \\
\frac{\delta-\beta}{\delta-1}\lambda_3^1 + \lambda_2^2 + \frac{1}{1-\delta}\lambda_3^2 + \lambda_4^2 &\leq 1 \text{ if } \alpha \geq \delta \\
\frac{\beta-\gamma}{\gamma-1}\lambda_3^1 + \frac{\alpha-\gamma}{\gamma-1}\lambda_4^1 + \frac{1}{\gamma-1}(\lambda_3^2 + \lambda_4^2) &\leq -1 \\
\frac{\delta-\beta}{\delta-1}\lambda_3^1 + \frac{\delta-\alpha}{\delta-1}\lambda_4^1 + \frac{1}{1-\delta}(\lambda_3^2 + \lambda_4^2) &\leq 1
\end{aligned}$$

Case 12: $p_1^2 < p_1^1 < p_3^2, p_1^2 < p_2^1 < p_3^2, p_3^1 > p_3^2, p_1^2 < p_4^1 < p_3^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	0	0	1	0	α	0	$1-\alpha$
0	0	0	1	α	0	0	$1-\alpha$
0	0	0	1	α	0	$1-\alpha$	0
0	0	β	$1-\beta$	0	0	0	1
0	0	β	$1-\beta$	0	0	1	0
0	1	0	0	0	γ	0	$1-\gamma$
0	1	0	0	γ	0	0	$1-\gamma$
0	1	0	0	γ	0	γ	0
δ	0	$1-\delta$	0	0	0	0	1
δ	0	$1-\delta$	0	0	0	1	0
1	0	0	0	0	ϵ	0	$1-\epsilon$
1	0	0	0	ϵ	0	0	$1-\epsilon$
1	0	0	0	ϵ	0	$1-\epsilon$	0

$$
\begin{aligned}
\frac{\gamma}{(\gamma-1)(\delta-1)}\lambda_3^1 + \frac{\beta\gamma+\gamma\delta-\gamma}{(\beta-1)(\gamma-1)(\delta-1)}\lambda_4^1 + \frac{1}{\gamma-1}(\lambda_3^2+\lambda_4^2) &\leq -1 \\
\text{if } \gamma(\beta+\delta-1) \geq (\beta-1)(\delta-1)(\alpha-\gamma) & \\
\lambda_2^2 - \frac{\gamma}{1-\gamma}\lambda_4^2 &\leq 0 \\
\frac{\alpha}{(\alpha-1)\beta}\lambda_3^1 + \lambda_2^2 + \frac{1}{1-\alpha}\lambda_3^2 + \lambda_4^2 &\leq 1 \text{ if } \delta \leq 1-\beta \\
\frac{\alpha\beta-\beta\gamma-\alpha}{\beta(\gamma-1)}\lambda_3^1 + \frac{\alpha-\gamma}{\gamma-1}\lambda_4^1 + \frac{1}{\gamma-1}(\lambda_3^2+\lambda_4^2) &\leq -1
\end{aligned}
$$

Case 13: $p_1^2 < p_1^1 < p_3^2, p_1^2 < p_2^1 < p_3^2, p_3^1 > p_3^2, p_4^1 > p_3^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	α	0	$1-\alpha$	0	0	0	1
0	α	0	$1-\alpha$	0	0	1	0
0	1	0	0	0	β	0	$1-\beta$
0	1	0	0	β	0	0	$1-\beta$
0	1	0	0	β	0	$1-\beta$	0
γ	0	0	$1-\gamma$	0	0	0	1
γ	0	0	$1-\gamma$	0	0	1	0
δ	0	$1-\delta$	0	0	0	0	1
δ	0	$1-\delta$	0	0	0	1	0
1	0	0	0	0	ϵ	0	$1-\epsilon$
1	0	0	0	ϵ	0	0	$1-\epsilon$
1	0	0	0	ϵ	0	$1-\epsilon$	0

$$\begin{aligned}
\frac{\beta}{(\delta-1)(\beta-1)}\lambda_3^1 + \frac{\beta}{(\alpha-1)(\beta-1)}\lambda_4^1 - \frac{1}{1-\beta}(\lambda_3^2+\lambda_4^2) &\leq -1 \\
\lambda_2^2 - \frac{\beta}{1-\beta}\lambda_4^2 &\leq 0 \\
-\frac{\epsilon}{(\delta-1)(\epsilon-1)}\lambda_3^1 - \frac{\epsilon}{(\gamma-1)(\epsilon-1)}\lambda_4^1 + \lambda_2^2 + \frac{1}{1-\epsilon}\lambda_3^2 + \lambda_4^2 &\leq 1 \\
-\frac{\epsilon}{(\delta-1)(\epsilon-1)}\lambda_3^1 - \frac{\epsilon}{(\gamma-1)(\epsilon-1)}\lambda_4^1 + \frac{1}{1-\epsilon}(\lambda_3^2+\lambda_4^2) &\leq 1
\end{aligned}$$

Case 14: $p_1^1 > p_3^2, p_2^1 < p_1^2, p_4^1 < p_1^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	0	α	$1-\alpha$	0	1	0	0
0	0	α	$1-\alpha$	1	0	0	0
0	0	β	$1-\beta$	0	0	0	1
0	0	β	$1-\beta$	0	0	1	0
γ	0	0	$1-\gamma$	0	1	0	0
γ	0	0	$1-\gamma$	1	0	0	0
δ	$1-\delta$	0	0	0	1	0	0
δ	$1-\delta$	0	0	1	0	0	0
ϵ	0	0	$1-\epsilon$	0	0	0	1
ϵ	0	0	$1-\epsilon$	0	0	1	0
ζ	$1-\zeta$	0	0	0	0	0	1
ζ	$1-\zeta$	0	0	0	0	1	0

$$\begin{aligned}
\frac{1-\alpha}{\alpha}\lambda_3^1 - \lambda_4^1 - \frac{\beta-\alpha}{\alpha}(\lambda_3^2+\lambda_4^2) &\leq 0 \\
\lambda_3^1 - \frac{\beta}{1-\beta}\lambda_4^1 &\leq 0
\end{aligned}$$

Case 15: $p_1^1 > p_3^2, p_2^1 < p_1^2, p_1^2 < p_4^1 < p_3^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	0	0	1	0	α	0	$1-\alpha$
0	0	0	1	α	0	0	$1-\alpha$
0	0	0	1	α	0	$1-\alpha$	0
0	0	β	$1-\beta$	0	0	0	1
0	0	β	$1-\beta$	0	0	1	0
0	γ	0	$1-\gamma$	0	1	0	0
0	γ	0	$1-\gamma$	1	0	0	0
δ	0	0	$1-\delta$	0	0	0	1
δ	0	0	$1-\delta$	0	0	1	0
ϵ	$1-\epsilon$	0	0	0	1	0	0
ϵ	$1-\epsilon$	0	0	1	0	0	0
ζ	$1-\zeta$	0	0	0	0	0	1
ζ	$1-\zeta$	0	0	0	0	1	0

$$\begin{aligned}
\frac{1-\beta}{\beta}\lambda_3^1 - \lambda_4^1 &\leq 0 \\
\frac{1}{\beta}\lambda_3^1 - (\lambda_3^2 + \lambda_4^2) &\leq 0 \\
\frac{\alpha\delta - \alpha\beta}{\beta(\alpha\delta - \alpha - \delta)}\lambda_3^1 - \frac{\alpha}{\alpha\delta - \alpha - \delta}\lambda_4^1 + \frac{\delta}{\alpha + \delta - \alpha\delta}(\lambda_3^2 + \lambda_4^2) &\leq 1 \\
\frac{\beta - \alpha\beta + \alpha\gamma}{(\alpha - 1)\beta(\gamma - 1)}\lambda_3^1 + \frac{1}{1-\gamma}\lambda_4^1 - \frac{\gamma}{(\alpha - 1)(\gamma - 1)}(\lambda_3^2 + \lambda_4^2) &\leq 1
\end{aligned}$$

$$\frac{\alpha\delta - \alpha\beta}{\beta(\alpha\delta - \alpha - \delta)}\lambda_3^1 - \frac{\alpha}{\alpha\delta - \alpha - \delta}\lambda_4^1 + \frac{\alpha\delta - \alpha\gamma - \delta}{\alpha\delta - \alpha - \delta}\lambda_2^2$$
$$+\frac{\delta}{\alpha + \delta - \alpha\delta}\lambda_3^2 + \frac{\alpha^2(\delta - \gamma) + \delta - 2\alpha\delta}{(\alpha - 1)(\alpha\delta - \alpha - \delta)}\lambda_4^2 \leq 1$$

Case 16: $p_1^1 > p_3^2, p_2^1 < p_1^2, p_4^1 > p_3^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	α	0	$1-\alpha$	0	0	0	1
0	α	0	$1-\alpha$	0	0	1	0
0	β	0	$1-\beta$	0	1	0	0
0	β	0	$1-\beta$	1	0	0	0
γ	$1-\gamma$	0	0	0	1	0	0
γ	$1-\gamma$	0	0	1	0	0	0
δ	$1-\delta$	0	0	0	0	0	1
δ	$1-\delta$	0	0	0	0	1	0

$$\lambda_3^1 \leq 0$$
$$\frac{1}{1-\beta}\lambda_4^1 + \frac{\beta - \alpha}{\beta - 1}(\lambda_3^2 + \lambda_4^2) \leq 1$$

Case 17: $p_1^1 > p_3^2, p_1^2 < p_2^1 < p_3^2, p_1^2 < p_4^1 < p_3^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	0	0	1	0	α	0	$1-\alpha$
0	0	0	1	α	0	0	$1-\alpha$
0	0	0	1	α	0	$1-\alpha$	0
0	0	β	$1-\beta$	0	0	0	1
0	0	β	$1-\beta$	0	0	1	0
0	1	0	0	0	γ	0	$1-\gamma$
0	1	0	0	γ	0	0	$1-\gamma$
0	1	0	0	γ	0	$1-\gamma$	0
δ	0	0	$1-\delta$	0	0	0	1
δ	0	0	$1-\delta$	0	0	1	0
ϵ	$1-\epsilon$	0	0	0	0	0	1
ϵ	$1-\epsilon$	0	0	0	0	1	0

$$\begin{aligned}
\frac{1}{\beta}\lambda_3^1 - \frac{1}{1-\beta}\lambda_4^1 &\leq 0\\
\lambda_2^2 - \frac{\gamma}{1-\gamma}\lambda_4^2 &\leq 0\\
\frac{\alpha\beta-\alpha-\beta\gamma}{\beta(\gamma-1)}\lambda_3^1 + \frac{\alpha-\gamma}{\gamma-1}\lambda_4^1 + \frac{1}{\gamma-1}(\lambda_3^2+\lambda_4^2) &\leq -1\\
\frac{\alpha\delta-\alpha\beta}{\beta(\alpha\delta-\alpha-\delta)}\lambda_3^1 - \frac{\alpha}{\alpha\delta-\alpha-\delta}\lambda_4^1 + \frac{\delta}{\alpha+\delta-\alpha\delta}(\lambda_3^2+\lambda_4^2) &\leq 1
\end{aligned}$$

$$\frac{\alpha\delta-\alpha\beta}{\beta(\alpha\delta-\alpha-\delta)}\lambda_3^1 - \frac{\alpha}{\alpha\delta-\alpha-\delta}\lambda_4^1 + \frac{1}{\gamma}(1-(1-\gamma)(1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)}))\lambda_2^2 +$$
$$\frac{\delta}{\alpha+\delta-\alpha\delta}\lambda_3^2 + (1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)})\lambda_4^2 \leq 1$$

Case 18: $p_1^1 > p_3^2, p_1^2 < p_2^1 < p_3^2, p_4^1 > p_3^2$

Ingoing pipe				Outgoing pipe			
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2
0	α	0	$1-\alpha$	0	0	0	1
0	α	0	$1-\alpha$	0	0	1	0
0	1	0	0	0	β	0	$1-\beta$
0	1	0	0	β	0	0	$1-\beta$
0	1	0	0	β	0	$1-\beta$	0
γ	$1-\gamma$	0	0	0	0	0	1
γ	$1-\gamma$	0	0	0	0	1	0

$$\lambda_3^1 \leq 0$$
$$\begin{aligned}
\frac{\beta}{(\alpha-1)(\beta-1)}\lambda_4^1 + \frac{1}{\beta-1}(\lambda_3^2+\lambda_4^2) &\leq -1\\
\frac{1}{\beta}\lambda_2^2 - \frac{1}{1-\beta}\lambda_4^2 &\leq 0
\end{aligned}$$

Lemma 46 *The inequalities we have given in Cases* $1, 2, \dots, 18$ *in the situation of Figure A.5 are indeed valid inequalities (facets) of* P_Δ *in this special situation.*

Proof. The idea behind the proofs is principle the same as in the previous proofs. Because of this we only give an interesting example and omit the other tiring and long winded calculations. Via the vertices of the polyhedra it can be calculated that the inequalities are indeed valid inequalities (even facets). We have only proven that all inequalities are valid which is sufficient for our lifting algorithms since we only need some special kinds of valid inequalities.
Let us show the inequality

$$\frac{\alpha\delta-\alpha\beta}{\beta(\alpha\delta-\alpha-\delta)}\lambda_3^1 - \frac{\alpha}{\alpha\delta-\alpha-\delta}\lambda_4^1 + \frac{1}{\gamma}(1-(1-\gamma)(1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)}))\lambda_2^2 +$$
$$\frac{\delta}{\alpha+\delta-\alpha\delta}\lambda_3^2 + (1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)})\lambda_4^2 \leq 1$$

in Case 17. We remember the vertices of the polyhedron P_Δ in this case:

Ingoing pipe				Outgoing pipe				Nr.
λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_1^2	λ_2^2	λ_3^2	λ_4^2	
0	0	0	1	0	α	0	$1-\alpha$	1
0	0	0	1	α	0	0	$1-\alpha$	2
0	0	0	1	α	0	$1-\alpha$	0	3
0	0	β	$1-\beta$	0	0	0	1	4
0	0	β	$1-\beta$	0	0	1	0	5
0	1	0	0	0	γ	0	$1-\gamma$	6
0	1	0	0	γ	0	0	$1-\gamma$	7
0	1	0	0	γ	0	$1-\gamma$	0	8
δ	0	0	$1-\delta$	0	0	0	1	9
δ	0	0	$1-\delta$	0	0	1	0	10
ϵ	$1-\epsilon$	0	0	0	0	0	1	11
ϵ	$1-\epsilon$	0	0	0	0	1	0	12

The calculations for the points $3, 5, 6, 7, 8, 10, 11, 12$ are almost trivial or a consequence of the calculation of the other points and so we omit them.

Calculation for point 1:

$$-\frac{\alpha}{\alpha\delta-\alpha-\delta}+\frac{\alpha}{\gamma}(1-(1-\gamma)(1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)}))+(1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)})(1-\alpha)=$$

$$-\frac{\alpha}{\alpha\delta-\alpha-\delta}+\frac{\alpha}{\gamma}+(1-\alpha-\frac{\alpha}{\gamma}(1-\alpha))(1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)})=$$

$$-\frac{\alpha}{\alpha\delta-\alpha-\delta}+\frac{\alpha}{\gamma}+\frac{\gamma-\alpha}{\gamma}+\frac{\gamma-\alpha}{\gamma}\,\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)}=\frac{\gamma}{\gamma}=1.$$

Calculation for point 2:
In this case obviously holds $\gamma>\alpha$, since $p_2^1<p_4^1$. Because of $0<\alpha<1, 0<\gamma<1, 0<\delta<1$ we get $\alpha\delta-\alpha-\delta<0$ and so we get

$$\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)}<0$$

and thus

$$(1-\gamma)(1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)})<1$$

and so we get

$$\frac{\alpha}{\gamma}(1-(1-\gamma)(1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)}))>0.$$

Now we calculate (consider the calculation for point 1):

$$-\frac{\alpha}{\alpha\delta-\alpha-\delta}+(1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)})(1-\alpha)<$$

$$-\frac{\alpha}{\alpha\delta-\alpha-\delta}+\frac{\alpha}{\gamma}(1-(1-\gamma)(1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)}))+(1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)})(1-\alpha)=1.$$

Calculation for point 4:
Because of $\gamma>\alpha$ we get $\delta(\gamma-\alpha)+\alpha>0$ and thus

$$\delta\gamma-\delta\alpha-\gamma+\alpha>-\gamma\Rightarrow\alpha(\delta-1)(\gamma-\alpha)+\alpha\gamma>0.$$

Now we can calculate the value of the inequality for point 4:

$$\frac{\alpha\delta-\alpha\beta}{\beta(\alpha\delta-\alpha-\delta)}\beta-\frac{\alpha}{\alpha\delta-\alpha-\delta}(1-\beta)+1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)}=$$

$$\frac{\alpha(\delta-1)}{\alpha\delta-\alpha-\delta}+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)}+1=\frac{\alpha(\delta-1)(\gamma-\alpha)+\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)}+1<1$$

because of our preliminary calculation.

Calculation for point 9:

$$-\frac{\alpha}{\alpha\delta-\alpha-\delta}(1-\delta)+1+\frac{\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)}=\frac{(\alpha\delta-\alpha)(\gamma-\alpha)+\alpha\gamma}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)}+1$$
$$=\frac{\alpha(\alpha+\delta(\gamma-\alpha))}{(\gamma-\alpha)(\alpha\delta-\alpha-\delta)}+1<1.$$

The inequality is fulfiled at equality at points $1,3,5,6,10$. So the inequality is valid. With an easy calculation we can calculate the dimension of the polyhedron and in this way we can test whether we indeed have found a facet. □

We add the following

Remark 47 *Even in the considered situations the calculated inequalities are not sufficient in order to separate all LP-solutions that do not fulfil the triangle conditions. From this fact we conclude that in the case that the valid inequalities are indeed facets we can not guarantee to calculate the complete description of $P_\triangle$ even in this small cases.*

Let us consider Case 2 for Figure A.5 with

$$p^1_{out}=\begin{pmatrix}25\\20\\35\\32\end{pmatrix}, p^2_{in}=\begin{pmatrix}30\\30\\40\\40\end{pmatrix}$$

Let the calculated LP-solution be

$$\begin{pmatrix}0\\0\\1\\0\\\frac{1}{4}\\\frac{1}{4}\\\frac{1}{4}\\\frac{1}{4}\end{pmatrix}.$$

It is easy to see that the calculated inequalities do not cut off this point since the calculated valid inequalities are

$$\begin{aligned}
-10\lambda^1_3-7\lambda^1_4+10x^2_3+10\lambda^2_4 &\leq -5\\
\lambda^2_2+2\lambda^2_3+\lambda^2_4 &\leq 1\\
-3\lambda^1_3+2\lambda^1_4+10\lambda^2_3+2\lambda^2_4 &\leq 2\\
10\lambda^1_3+6\lambda^1_4-10\lambda^2_3-10\lambda^2_4 &\leq 5\\
15\lambda^1_3+12\lambda^1_4-10\lambda^2_3-10\lambda^2_4 &\leq 10.
\end{aligned}$$

Let the calculated LP-solution be

$$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \frac{1}{2} \\ \frac{1}{4} \\ \frac{1}{4} \end{pmatrix}.$$

In this case it is easy to see that the calculated inequalities cut off this point because we get in the second inequality

$$\frac{1}{2} + 2 \cdot \frac{1}{4} + \frac{1}{4} > 1.$$

For the use of separation algorithms as described in Chapter 6 we have to mention the problem that every more complicated discretisation graph induces new classes of valid inequalities and facets. These new classes of inequalities of course cannot be calculated via lifting from known inequalities of smaller discretisation graphs.

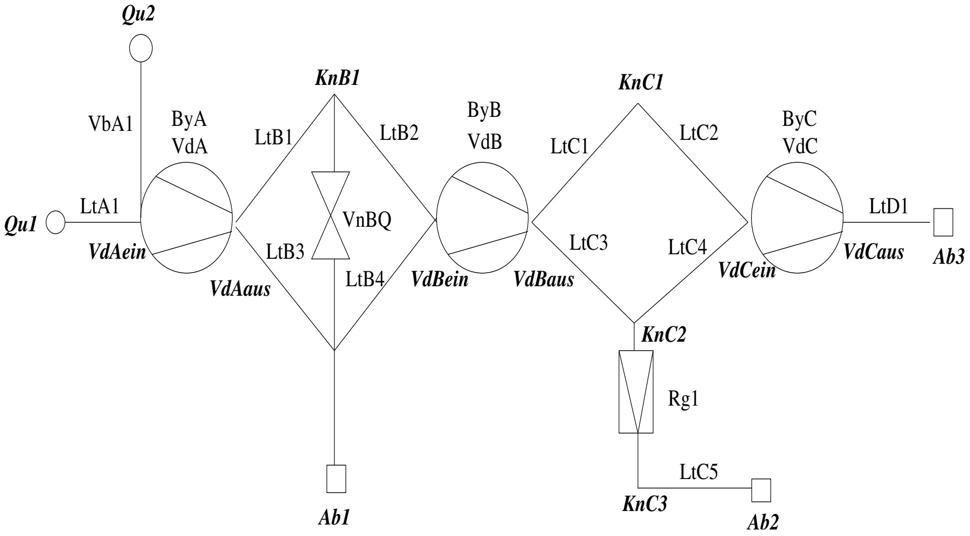

Figure A.6: Nodes and segments of used test model

A.3 Complete Data of Test Models

Figure A.6 shows a small gas network that we used for the first test calculations. In the following tables we give the data of all segments and nodes.

Pipes (length L, diameter D, pipe roughness k):

	L	D	k
LtA1	110000	1.1	0.00001
LtB1	100000	0.95	0.000015
LtB2	60000	0.95	0.000015
LtB3	100000	0.9	0.000015
LtB4	60000	0.9	0.000015
LtC1	100000	0.9	0.000012
LtC2	50000	0.9	0.000012
LtC3	100000	0.85	0.000012
LtC4	50000	0.85	0.000012
LtC5	70000	0.5	0.00005
LtD1	120000	1.1	0.00001

Nodes (minimal pressure p_{min}, maximal pressure p_{max}):

	p_{min}	p_{max}
Qu1	60	69
Qu2	45	58
Ab1	45	70
Ab2	45	49
Ab3	50	60
VdAein	45	70
VdAV	45	70
KnB1	45	70
KnB2	45	70
VdBein	45	70
VdBaus	45	70
KnC1	45	70
KnC2	45	70
KnC3	45	70
VdCein	45	70
VdCV	45	70
VdCaus	45	70

Compressors (adiabatic efficiency η_{ad}, specific heat rate b, minimal power N_{min}, maximal power N_{max}):

	η_{ad}	b	N_{min}	N_{max}
VdA	0.8	12000	10000	60000
VdB	0.75	11700	3000	45000
VdC	0.65	12600	3000	45000

Segments (minimal gas flow q_{min}, maximal gas flow q_{max}, here given in the dimension $[1000m^3/h]$):

	q_{min}	q_{max}
LtA1	0	2500
LtB1	0	2000
LtB2	0	2000
LtB3	0	2000
LtB4	0	2000
LtC1	0	2000
LtC2	0	2000
LtC3	0	2000
LtC4	0	2000
LtC5	0	500
LtD1	0	2000
VbA1	300	300
VbB1	0	200
VdA	0	3000
VdB	0	3000
VdC	0	3000
ByA	0	3000
ByB	0	3000
ByC	0	3000
VnBQ	-200	400
Rg1	0	60

Minimal gas flow q_{min} and maximal gas flow q_{max} for gas delivering nodes ($Qu1, Qu2$) and sinks ($Ab1, Ab2, Ab3$), here given in the dimension $[1000m^3/h]$:

	q_{min}	q_{max}
Qu1	1000	2500
Qu2	300	300
Ab1	50	50
Ab2	60	60
Ab3	1650	1650

Control valve (minimal pressure reduction $p_{dp_{min}}$, maximal pressure reduction $p_{dp_{max}}$):

	$p_{dp_{min}}$	$p_{dp_{max}}$
Rg1	1	10

Summary of all important non-vanishing solution-values:

q_{LtA1}: 1469.355096
q_{LtB1}: 948.531826
q_{LtB2}: 914.869159
q_{LtB3}: 817.121561
q_{LtB4}: 800.784229
q_{LtC1}: 910.052683
q_{LtC2}: 910.052683
q_{LtC3}: 802.503450
q_{LtC4}: 742.503450
q_{LtC5}: 60.000000
q_{LtD1}: 1650.000000
q_{VbA1}: 300.000000
q_{VnBQ}: 33.662668
s_{VnBQ}: 1.000000
q_{Rg1}: 60.000000
s_{Rg1}: 1.000000
q_{VdA}: 1769.355096
N_{VdA}: 11037.569472
f_{VdA}: 3.701709
s_{VdA}: 1.000000
q_{VdB}: 1715.653388
N_{VdB}: 9472.041959
f_{VdB}: 3.097255
s_{VdB}: 1.000000
q_{VdC}: 1652.556133
N_{VdC}: 7258.807864
f_{VdC}: 2.556133
s_{VdC}: 1.000000
q_{Qu1}: 1469.355096
p_{Qu1}: 70.013250
q_{Qu2}: 300.000000
p_{Qu2}: 59.013250
q_{Ab1}: 50.000000
p_{Ab1}: 64.150565

q_{Ab2}: 60.000000
p_{Ab2}: 50.013250
q_{Ab3}: 1650.000000
p_{Ab3}: 51.013250
p_{VdAein}: 59.013250
p_{VdAaus}: 71.013250
p_{KnB1}: 64.150565
p_{VdBein}: 59.916834
p_{VdBaus}: 69.865576
p_{KnC1}: 61.465123
p_{KnC2}: 60.925951
p_{KnC3}: 50.925951
p_{VdCein}: 56.733354
p_{VdCaus}: 63.980631

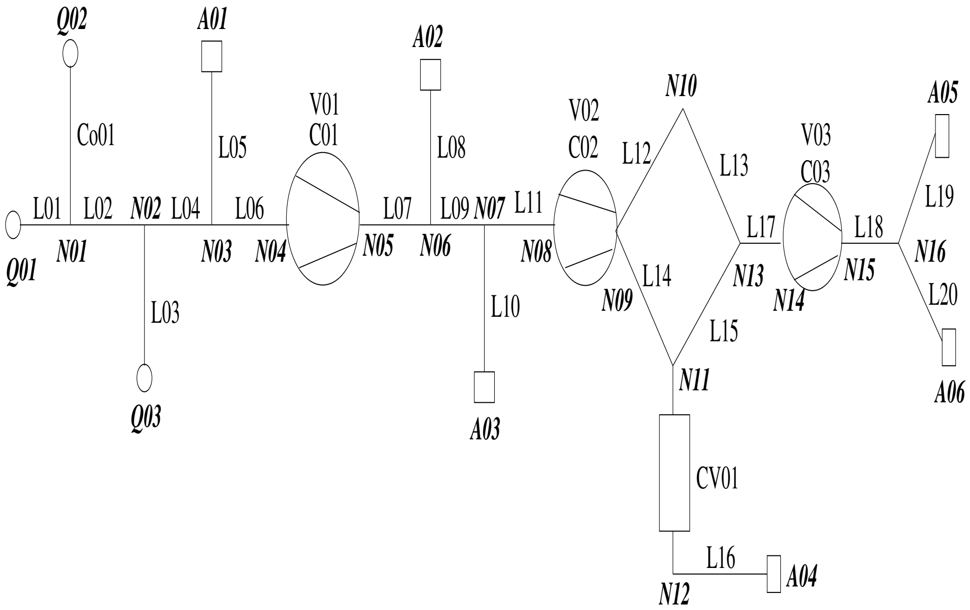

Figure A.7: Nodes and segments of modified and heightened test model

Figure A.7 shows a modified and little heightended version of the used test model. Here we give the data of this second model and give show how the solution time depends on the initional conditions. First we give the data of the standard situation:

Pipes (length L, diameter D, pipe roughness k):

	L	D	k
L01	50000	1.1	0.00001
L02	50000	1.1	0.00001
L03	50000	1.1	0.00001
L04	50000	1.1	0.00001
L05	50000	1.1	0.00001
L06	50000	1.1	0.00001
L07	30000	0.9	0.00001
L08	30000	0.9	0.00001
L09	30000	0.9	0.00001
L10	30000	0.9	0.00001
L11	30000	0.9	0.00001
L12	100000	0.9	0.000012
L13	50000	0.9	0.000012
L14	100000	0.85	0.000012
L15	50000	0.85	0.000012
L16	30000	0.5	0.00001
L17	50000	0.9	0.000012
L18	50000	1.1	0.00001
L19	50000	1.1	0.00001
L20	50000	1.1	0.00001

Nodes (minimal pressure p_{min}, maximal pressure p_{max}):

	p_{min}	p_{max}
Q01	60	69
N01	40	70
Q02	45	68
N02	40	70
Q03	40	70
N03	40	70
A01	40	70
N04	40	70
N05	40	70
N06	40	70
A02	40	70
N07	40	70
A03	40	70
N08	40	70
N09	40	70
N10	40	70
N11	40	70
N12	40	70
A04	40	70
N13	40	70
N14	40	70
N15	40	70
N16	40	70
A05	40	60
A06	40	60

Compressors (adiabatic efficiency η_{ad}, specific heat rate b, minimal power N_{min}, maximal power N_{max}):

	η_{ad}	b	N_{min}	N_{max}
C01	0.8	12000	3000	60000
C02	0.8	11700	3000	45000
C03	0.8	12600	3000	45000

Segments (minimal gas flow q_{min}, maximal gas flow q_{max}, here given in the dimension $[1000m^3/h]$):

	q_{min}	q_{max}
L01	0	2500
C001	0	3000
L02	0	3000
L03	0	3000
L04	0	3000
L05	0	3000
L06	0	3000
C01	0	3000
V01	0	3000
L07	0	3000
L08	0	3000
L09	0	3000
L10	0	3000
L11	0	3000
C02	0	3000
V02	0	3000
L12	0	3000
L13	0	3000
L14	0	3000
L15	0	3000
CV01	0	3000
L16	0	3000
L17	0	3000
C03	0	3000
V03	0	3000
L18	0	2000
L19	0	3000
L20	0	3000

Minimal gas flow q_{min} and maximal gas flow q_{max} for gas delivering nodes ($Qu1, Qu2$) and sinks ($Ab1, Ab2, Ab3$), here given in the dimension $[1000m^3/h]$:

	q_{min}	q_{max}
Q01	1000	2500
Q02	500	2000
Q03	100	2000
A01	50	1000
A02	200	1000
A03	250	1000
A04	50	300
A05	600	800
A06	500	800

Control valve (minimal pressure reduction $p_{dp_{min}}$, maximal pressure reduction $p_{dp_{max}}$):

	$p_{dp_{min}}$	$p_{dp_{max}}$
CV01	1	10

Bibliography

[1] I.R. de Farias A.B. Kehe and G.L.Nemhauser. Models for representing piecewise linear cost functions. Technical report, CORE, 2002.

[2] Ruhrgas AG. WWW Page: http://www.ruhrgas.com.

[3] Ruhrgas AG. Nonpublished.

[4] E. M. L. Beale and J. A. Tomlin. Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. *Proceedings of the Fith International Conference on Operations Research, Tavistock Publications*, pages 447 – 454, 1970.

[5] I.N. Bronstein and K.A. Semendjajew. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Thun und Frankfurt/Main, 1987.

[6] T. Christof and A. Löbel. *PORTA: POlyhedron Representation Transformation Algorithm, Version 1.3*. Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2000.

[7] Vašek Chvátal. *Linear Programming*. W. H. Freeman and Company, 1983.

[8] G.B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. In T.C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 330 – 335. Wiley, New York, 1951.

[9] Verband der schweizerischen Gasindustrie. WWW Page: http://www.erdgas.ch.

[10] K. Ehrhardt. Comments to Tto-Basisdoc. Preprint, Konrad-Zuse-Zentrum Berlin, 2001.

[11] K. Ehrhardt and M. Steinbach. Betriebskostenminimierung für Erdgas-Transportnetze. Technical report, Konrad Zuse Zentrum für Informationstechnik, Berlin, 2002.

[12] C.E. Ferreira, A. Martin, and R. Weismantel. Solving multiple knapsack problems by cutting planes. *SIAM Journal on Optimization*, 6:858 – 877, 1996.

[13] A. Fügenschuh and A. Martin. Computational integer programming and cutting planes. Preprint 2221, Technical University Darmstadt, 2002.

[14] V.N. Gopal. Techniques to optimize fuel and compressor combination selection. In *American Gas Association Transmission Conference*, 1979.

[15] M. Grötschel, A. Martin, and R. Weismantel. Packing Steiner trees: Polyhedral investigations. *Mathematical Programming*, 72:101 – 123, 1996.

[16] P. Hackländer. *Integrierte Betriebsplanung von Gasversorgungsunternehmen*. PhD thesis.

[17] E.L. Johson I.R. de Farias, Jr. and G.L.Nemhauser. A generalized assignment problem with special ordered sets: a polyhedral approach. Technical report, Preprint University of New York at Buffalo, 2000.

[18] Jr. I.R. de Farias. Branch-and-cut for combinatorial optimization problems without auxiliary binary variables. Technical report, Preprint University of New York at Buffalo, 2000.

[19] T. Jenicek. Steady-state optimization of gas transport. In *Proceedings of the 2nd International Workshop SIMONE on Innovative Approaches to Modelling and Optimal Control of Large Scale Pipeline Networks*, September 1993.

[20] J. Kallrath. *Gemischt-Ganzzahlige Optimierung: Modellierung in der Praxis*. Vieweg, 2002.

[21] C. Kelling, K. Reith, and E. Sekirnjak. A practical approach to transient technical optimization for gas networks. Research note, Ruhrgas AG, PSI AG, 2000.

[22] L.G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademiia Nauk SSSR*, 244:1093 – 1096, 1979. (translated in Soviet Mathematics Doklady **20**, 191 – 194, 1979.

[23] T. Koch, A. Martin, and S. Voß. *SteinLib: An Updated Library on Steiner Tree Problems in Graphs*, pages 285–325. Kluwer, 2001.

[24] J. Králik. Compressor stations in SIMONE. In *Proceedings of the 2nd International Workshop SIMONE on Innovative Approaches to Modelling and Optimal Control of Large Scale Pipeline Networks*, September 1993.

[25] ILOG CPLEX 7.0: Reference Manual.

[26] ILOG CPLEX 8.0: Reference Manual.

[27] H. Marchand, A. Martin, R. Weismantel, and L.A. Wolsey. Cutting planes in integer and mixed integer programming. Technical Report CORE DP9953, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 1999.

[28] H. Marchand, A. Martin, R. Weismantel, and L.A. Wolsey. Cutting planes in integer and mixed integer programming. *Discrete Applied Mathematics*, 123/124:391 – 440, 2002.

[29] A. Martin. *Packen von Steinerbäumen: Polyedrische Studien und Anwendungen*. PhD thesis, Technische Universität Berlin, 1992.

[30] A. Martin. Integer programs with block structure. Technical Report Preprint SC 99-03, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1999.

[31] A. Martin. Large scale optimization. Preprint Nr. 2166, Darmstadt University of Technology, Department of Mathematics, August 2001. To appear in EOLSS.

[32] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1988.

[33] M.W. Padberg. *Linear Optimization and Extensions*. Springer, 1995.

[34] Christos H. Papadimitriou. *Combinatorial Optimization*. Prentice-Hall, 1982.

[35] K.F. Pratt and J.G. Wilson. Optimization of operation of gas transmission systems. *Transactions of the Institute of Measurement and Control*, 6(4):261 – 269, October 1984.

[36] Carter R. Pipeline optimization: Dynamic programming after 30 years. In *Pipeline Simultaion Interest Group*, URL: www.psig.com, 1998.

[37] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chichester, 1986.

[38] E. Sekirnjak. *Mixed Integer Optimization for Gas Transmission and Distribution Systems*. INFORMS Meeting, Seattle, October 1998. Vortragsmanuskript.

[39] E. Sekirnjak. *Transiente Technische Optimierung (TTO-Prototyp)*. PSI Berlin, 1999. Vertrauliche Dokumentation.

[40] E. Sekirnjak. *Stationäre Leitungshydraulik für lange Segmente*. PSI Berlin, 2002. Vertrauliche Dokumentation.

[41] E. Sekirnjak. *Transiente Leitungsmodelle für lange Segmente und Perioden*. PSI Berlin, 2002. Vertrauliche Dokumentation.

[42] A. Sirleschtov. Im Wettbewerb: Mit der bevorstehenden Liberalisierung des Gasmarktes beginnt der Kampf um die Kunden. In *Der Tagesspiegel*, 29. Januar 2000.

[43] Yves Smeers and Daniel De Wolf. The gas transmission problem solved by an extension of the simplex algorithm. *MANAGEMENT SCIENCE*, Vol. 46, Nr. 11:1454–1465, 2000.

[44] F. van Oostvoorn. European gas market developments - oportunities and threats -. In *Supplement to the IAEE/GEE Conference "Energy Markets, What's New?"*, pages 29 – 44, Berlin, 9-10 September 1998.

[45] Z. Vostrý. Transient optimization of gas transport and distribution. In *Proceedings of the 2nd International Workshop SIMONE on Innovative Approaches to Modelling and Optimal Control of Large Scale Pipeline Networks*, September 1993.

[46] M. Westphalen. *Anwendungen der stochastischen Optimierung im Stromhandel und Gastransport*. PhD thesis, Universität Duisburg-Essen, 2004.

[47] H.P. Williams. *Model building in mathematical programming*. Wiley, 1990.

[48] L.A. Wolsey. Faces of linear inequalities in 0-1 variables. *Mathematical Programming*, 8:165 – 178, 1975.

[49] L.A. Wolsey. *Integer Programming*. Wiley, 1998.

[50] S. Wright, M. Somani, and C. Ditzel. Compressor station optimization. In *Pipeline Simulation Interest Group*, Denver, Colorado, October 1998.

[51] H.I. Zimmer. Calculating optimum pipeline operations. In *American Gas Association Transmission Conference*, 1975.

[52] R.D. Zucker. *Fundamentals in gas dynamics*. Wiley, Hoboken, NJ, 2002.

Akademischer Werdegang

31.12.1973	Geboren in Rotenburg/Fulda
1980-1984	Besuch der Grundschule Bebra-Breitenbach
1984-1993	Besuch des Wigbert-Gymnasiums Hünfeld
1993	Abitur
10.1993-06.1999	Studium der Mathematik mit Nebenfach Chemie am Fachbereich Mathematik der Technischen Universität Darmstadt
06.1999	Diplom
08.1999-12.2000	Wissenschaftlicher Mitarbeiter am Lehrstuhl und Seminar für Allgemeine Betriebswirtschaftslehre, Organisation und Wirtschaftsinformatik der Universität Mannheim (Prof. Dr. F. Steffens) im Rahmen des Projekts: “Entwurf und Implementierung eines Metainformationssystems”
01.2001-	Wissenschaftlicher Mitarbeiter der Arbeitsgruppe“Diskrete Optimierung” des Fachbereiches Mathematik der Technischen Universität Darmstadt (Prof. Dr. A. Martin) im Rahmen des Projekts: ”Gemischt-Ganzzahlige Optimierung im operativen Gasmangement”

Erklärung

Hiermit erkläre ich, dass ich vorliegende Dissertation selbstständig und nur unter Zuhilfenahme der angegebenen Hilfsmittel und Quellen angerfertigt habe.

Darmstadt, den 24.Februar 2004,

Markus Möller.