Optimal Operation of Water Supply Networks by Mixed Integer Nonlinear Programming and Algebraic Methods

vom Fachbereich Mathematik
der Technischen Universität Darmstadt
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)
genehmigte Dissertation

Tag der Einreichung: 01.02.2019
Tag der mündlichen Prüfung: 16.04.2019

Gutachter:	Prof. Dr. Marc E. Pfetsch
	PD Dr. Raymond Hemmecke

von
Dipl.-Math. Wei Huang
aus Hubei (China)

Darmstadt, D 17
2019

Huang, Wei: Optimal Operation of Water Supply Networks by Mixed Integer Nonlinear Programming and Algebraic Methods
Darmstadt, Technische Universität Darmstadt
Jahr der Veröffentlichung der Dissertation auf TUprints: 2019
URN: urn:nbn:de:tuda-tuprints-86575
Tag der mündlichen Prüfung: 16.04.2019

For my family

Abstract

In this thesis we are dealing with the operative planning of water supply networks. The task of an operative planning is to create a pump and valve configuration such that the water requirement from consumers is fulfilled with necessary quality. An optimal operation corresponds to a configuration that minimizes the operation cost as well as potential water procurement cost.

There are different ways to handle this problem. We solve it as an optimization problem using mathematical programming.

On the one hand, the network problem contains some discrete variables, for example, the pump or valve status; on the other hand, nonlinearities and nonconvexities from physical behaviors make the mathematical model extremely difficult. We model the optimization problem as a mixed-integer nonlinear program (MINLP).

We choose MINLP solver SCIP, developed mainly at Zuse Insitute Berlin. We use two real-world instances provided by industrial partner Siemens and a further real-world instance obtained from the Department of Hydraulic Engineering of Tsinghua University.

In this thesis, we first show that our solver SCIP is able to solve the optimal operation problem to global optimality in a fixed point of time. However, for a daily operation which contains 24 coupled time periods (hours), "good" solutions are usually found rapidly, but the dual gap cannot verify the solution quality.

In a further chapter we show that a class of subnetworks which only contains pipes and consumers, can be simplified and the original nonlinear constraints can be replaced by few (or single) nonlinear constraints, without changing the feasible region. Computation shows that this simplification makes the MINLP easier to solve.

The algorithm which solves our nonconvex MINLP generates at every iteration a convex relaxation of the feasible region. A lot of theories and experiments showed that tighter convex relaxation is quite relevant for the branch-and-bound approach.

In the objective of our model, we have bivariate polynomial term with degree 3. Based on the default construction of convex relaxation, we want to find additional linear constrains ("valid cuts") to make the relaxation tighter. We investigate the graph of general polynomial functions over a polytope in general dimension and develop theory to describe the convex hull of the graph and to find halfspaces which contain the convex hull. After that we define "tight" halfspaces which denote the "efficient" halfspaces when forming the convex hull. For bivariate polynomial functions with degree 3, algorithms are designed to find such tight halfspaces. After adding these halfspaces (linear constraints) into the MINLP, computation shows that both primal and dual bound are definitively improved within the same time limit.

Zusammenfassung

In dieser Arbeit beschäftigen wir uns mit der operativen Planung von Wasserversorgungsnetzen. Die Aufgabe einer operativen Planung ist eine Pump- und Ventilkonfiguration zu erstellen, so dass der Wasserbedarf von Verbrauchern mit notwendiger Qualität erfüllt wird.

Ein optimaler Betrieb entspricht einer Konfiguration, welche die Betriebskosten sowie mögliche Wasserbeschaffungskosten minimiert. Es gibt verschiedene Möglichkeiten, dieses Problem zu lösen. Wir lösen es als ein Optimierungsproblem mit mathematischer Optimierung. Einerseits enthält das Netzwerkproblem einige diskrete Variablen, zum Beispiel der Pumpen- oder Ventilstatus; andererseits, machen Nichtlinearitäten und Nichtkonvexitäten aus physikalischem Verhalten das mathematische Modell extrem schwierig. Wir modellieren das Optimierungsproblem als ein gemischt-ganzzahliges nichtlineares Programm (MINLP).

Wir haben uns für den MINLP-Solver SCIP entschieden, der im Zuse Institut Berlin entwickelt wird. Wir verwenden zwei reale Instanzen bereitgestellt von dem Industriepartner Siemens und eine weitere reale Instanz versorgt von der Fakultät Hydraulic Engineering der Tsinghua-Universität.

Wir zeigen zuerst, dass unser Solver SCIP in der Lage ist, das optimale Planungsproblem der Wasserversorgungsnetzen in einem festen Zeitpunkt zu globaler Optimalität zu lösen. Allerdings können zwar *gute* Lösungen für den täglichen Betrieb, welcher 24 gekoppelte Zeiträume (Stunden) enthält, gefunden werden. Deren Qualität kann allerdings wegen der schwachen dualen Schranke nicht bestätigt werden.

In einem weiteren Kapitel zeigen wir, dass eine Klasse von Teilnetzen, welche nur Wasserrohren und Verbraucher enthalten, vereinfacht werden kann. Mathematisch zeigen wir, dass die ursprünglichen nichtlinearen Nebenbedingungen durch wenige (oder einzelne) Nebenbedingungen ersetzt werden können ohne den zulässigen Bereich zu ändern. Numerische Ergebnisse zeigen, dass diese Vereinfachung die MINLPs deutlich einfacher lösbar macht.

Der Algorithmus, welcher unser nichtkonvexes MINLP löst, erzeugt bei jeder Iteration eine konvexe Relaxation des zulässigen Bereichs. Viele Theorien und Experimente zeigten, dass eine engere konvexe Relaxation für den Branch-and-Bound-Ansatz ziemlich relevant ist.

In der Zielfunktion unseres Modells haben wir nichtlineare Funktionen in Form von bivariaten Polynomen mit Grad 3. Basierend auf der Standardkonstruktion der konvexen Relaxation wollen wir noch zusätzliche lineare Nebenbedingungen (*gültige Schnitte*) finden, um die Relaxation enger zu machen. Wir untersuchen den Graphen von allgemeinen Polynomfunktionen, der auf einem Polytop definiert ist, und entwickeln eine Theorie, um die konvexe Hülle des Graphen zu beschreiben und um Halbräume zu finden, welche die konvexe Hülle enthalten. Danach definieren wir *enge* Halbräume, die die *effizienten* Halbräume bei Darstellung der konvexen Hülle bezeichnen. Für bivariate Polynomfunktionen mit Grad 3 werden Algorithmen entwickelt, um solche engen Halbräume zu finden. Nach dem Hinzufügen solcher engen Halbräume (lineare Nebenbedingungen) in das MINLP, zeigen unsere weiteren numerischen Berechnungen, dass sowohl die primale als die duale Schranke innerhalb derselben Zeitlimite definitiv verbessert werden.

Acknowledgement

First of all, I would like to thank my supervisor PD. Dr. Raymond Hemmecke during the first phase of my doctoral studies at Technische Universität München in Munich and my supervisor Prof. Dr. Marc E. Pfetsch during the final phase of my doctoral studies in Darmstadt. I am very thankful for all the fruitful discussions, patience, encouragement and the support from you both.

Next, I want to thank Prof. Christopher T. Ryan for inviting me to visit the University of Chicago. With his supervision and encouragement I finished the theoretical part of Chapter 5. Furthermore I want to thank Prof. Jiahua Wei to support me for a research stay at the Tsinghua University. There I got advices and obtained a real-world instance to finish the computational part of Chapter 5.

Many thanks go to Dr. Christoph Moll and Dr. Harald Held from Siemens AG for their supervision, especially contributions from industrial point of view and preparing computational data.

In addition, I want to thank all (former) members of the work group "Modeling, Simulation and Optimization" from Siemens AG, of the research group "Applied Geometry and Discrete Mathematics" of TU München and of the research group "Optimization" of TU Darmstadt. Thank you for a wonderful working atmosphere and collaboration. In the final year in Darmstadt, the quality of the manuscript was improved significantly by the proofreading of Dr. Christopher Hojny, Andreas Schmitt, Oliver Habeck, Frederic Matter and Prof. Dr. Marc E. Pfetsch. Thank you very much for the effort.

Special thanks goes to the SCIP group, in particular to my former supervisor Dr. Ambros Gleixner, to Dr. Stefan Vigerske for the algorithmic support and to my supervisor Prof. Dr. Marc E. Pfetsch. I am also very grateful to Prof. Dr. Oliver Kolb for his support in the early phase of the thesis.

Last but not least, my biggest thanks are for my entire family, in particular my parents who support me since my birth, my parents in law for the love in the last years, my wife Di who supported me during my whole studies and our kid Yishi who accompanied almost the whole phase of my doctoral studies.

Contents

Chapter 1

Introduction

Water is one of the most important substances in our life. Clean drinking water is essential to human and other forms of life. Water also plays a significant role in economies, for it works as a solvent for many chemical substances, e.g., 70% of freshwater is consumed by agriculture.

Water treatment has been critical for a long time. Each country or city has its own requirements which determine treatment needs. A few hundred years ago, a good water supply system was even one of the marks of an advanced civilization.

We introduce in Section 1.1 the definition of water supply and water supply networks, while Section 1.2 contains details on the objective for optimal operation of water supply networks (OOWSN). A literature survey for related problems follows in Section 1.3. Afterwards, we give a short introduction to the mathematical method *mixed integer nonlinear programming* in Section 1.4. At the end there is an outline of the thesis. Parts of this chapter have been published in [Hua11].

1.1 Water supply networks

Water supply is a provision of public utilities, commercial organizations, communities etc. and usually supplied by a system of pipes and pumps. One of the best-known examples is our drinking water systems. In 2010, about 84% of the global population (6.74 billion people) had access to a piped water supply through house connections or through an improved water source other than in-house, including standpipes, “water kiosks”, protected springs and protected wells[1].

Water supply systems get water from a variety of locations, including groundwater, surface water (lakes and rivers), and from the sea through desalination. These water sources cannot be used directly in most cases. The water must be purified, e.g., disinfected through chlorination or sometimes fluoridated. Treated water then either flows by gravity or is pumped to reservoirs.

Another key concept in water supply systems is the water pressure. Water pressure varies in different locations of a distribution system. In poorly managed systems, water pressure can be so low as to result in only a trickle of water or so high that it leads to damaged plumbing

[1] Progress on Sanitation and Drinking-water: 2010 Update, UNICEF, WHO/UNICEF Joint Monitoring Programme for Water Supply and Sanitation

fixtures and waste of water. Pressure in an urban water system is typically maintained either by a tank serving an urban area, by pumping the water up into a tower and relying on gravity to maintain a constant pressure, or by pumps at the water treatment plant and repeater pump stations.

> A water supply system is infrastructure for the collection, transmission, treatment, storage, and distribution of water for homes, commercial establishments, industry, and irrigation, as well as for such public needs as fire fighting and street flushing. Of all municipal services, provision of potable water is perhaps the most vital. People depend on water for drinking, cooking, washing, carrying away wastes, and other domestic needs. Water supply systems must also meet requirements for public, commercial, and industrial activities. In all cases, the water must fulfill both quality and quantity requirements[2].

A water supply system is also a system of engineered hydraulic components which provide water supply. A water supply system typically includes:

1. Sources of raw water, e.g., from a lake, a river, or groundwater.
2. Water treatment facilities, which purify raw water to clean water, e.g., drinking water or industrial water.
3. Pipelines, which are used to transfer treated water.
4. Water storage facilities, such as reservoirs, water tanks, or water towers.
5. Additional water pressurizing components, such as pumping stations.
6. A network for distribution of water to the consumers.
7. Valves, which can be active, partially active or inactive.

The water supply network that this thesis deals with is a part of the overall larger water supply system. Firstly, we will assume that all water to be transferred is purified and treated. For our intended purposes, we need not know where the raw water comes from and how it is purified etc. The network can get water from reservoirs, some foreign water companies and sometimes even protected springs and protected wells. However, if raw water is supplied, we only add the purifying cost to its price.

Our water supply network contains the following six components:

1. Reservoirs, water companies, or protected springs and wells. These all contain treated water. Water from companies is usually more expensive, but nearer to some consumers or tanks.

[2] Encyclopedia Britannica. 2010. Encyclopedia Britannica Online. 13 Oct. 2010

2. Pump stations. A pump station contains at least one pump. Several pumps in the same pump station can be connected in parallel or in series. Serially connected pumps increase the pressure one after another, and pumps connected in parallel increase the total flow rate.

3. Tanks. Tanks are intermediate storages of water with limited capacity.

4. Consumers. Consumers are given with demand (flow rate) which is estimated stochastically or empirically, usually given in a short time period, e.g., one hour.

5. Valves. An inactive (turned off) valve shuts off a subnetwork, i.e., controls the flow direction. A partially active valve reduces the pressure between two connected components.

6. Pipelines. Water is transferred through pipelines, which experience pressure loss according to material, length, and diameter of pipes and according to the flow rate of water.

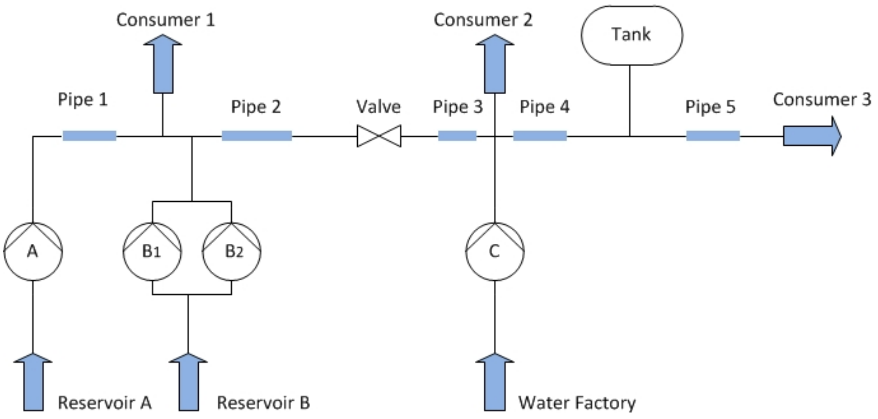


Figure 1.1: An example of a possible water supply network

In Figure 1.1 there is an example of a hypothetical water supply network. In this network, pump B_1 and pump B_2 have been connected in parallel and if at least one of these is active, water will be pumped from reservoir B. The price of water from our water factory may be more expensive than water from reservoirs A and B. But sometimes, it could be beneficial to pump more required water from the water factory rather than from reservoirs A or B. The demands in our consumers $1, 2, 3$ could deviate from the estimated amount, this may cause either oversupply or undersupply. Such cases can be handled with water tanks of flexible height.

The main task of operative planning of a water supply network is finding a pump configuration for all pumps in all pump stations so that the quantity requirements will be fulfilled. This problem can be treated as a multiobjective optimization problem. The next section gives more details for the multiple objectives.

1.2 Multiple objectives of optimal operation of water supply networks

There are several objectives for optimal operation of water supply networks:

1. Safe operation. The entire system should be safe. Operating components according to the manufacturer's recommendations should lead to less problems and malfunctions.

2. Full supply of consumers (robust supply). For a complete configuration, we should estimate the amount of demand for several short time periods in the future. But it is impossible to estimate them exactly. Thus we want to avoid situations in which water supply is scheduled from water tanks that are empty at the planned supply time. We also want to avoid situations in which tanks are overfilled due to overestimated consumer demand. In addition, we should also consider circumstances such as broken pumps and damaged pipes, under which the system as a whole must still function.

3. Minimal costs for water and energy. Pump configurations are executed by companies or public utilities. The more they pay for water and energy, the less the companies earn and the more financial problems the public utilities and communities have.

4. As few as possible pump switches. For technical reasons the pumps should not be switched too frequently. Switching too frequently reduces the life cycle and the efficiency of pumps. The efficiency of energy transformation from electrical energy to kinetic energy (flow rate) and potential energy (pressure) is extremely low during starting and ending.

1.3 Previous work

A systematic introduction to water supply and the distribution systems can be found in [WK07]. As described in Chapter 10 there, in the early days of water supply computer modeling, simulations were primarily used to solve design problems. At that time, operators preferred measuring pressures and flows in the field rather than working with complicated computer programs. Recent advances in software technology have made models more powerful and easier to use. As a result, operations personnel have accepted computer as a tool to help them in keeping the supply networks running smoothly.

From the literature point of view, a very early work with computer programs can be found in [WS73]. In this work, a detailed convex NLP was formulated and the Jacobian Differential Algorithm was developed to solve the NLP which is a generalized eliminating procedure and computationally feasible. The team implemented the algorithm with 25 subroutines. However, with the computational ability almost half of a century ago, very limited network size and constraint type can be handled.

A little more recent work for large-scale water supply networks can be found in [DLMY95]. The results there indicated that the combination of using an LP procedure and a graph algorithm is a very versatile tool for solving operation of large-scale water supply networks. Note that this work does not consider water pressure and pump scheduling. Meanwhile, a survey of research on optimal operation of water supply networks can be found in [ZS89].

Including optimal operation of water supply networks, there are also works with additional purposes, e.g., considering water quality [SM00] as well as considering preemptive priorities [BZBY08].

The most similar model to which we detail in Chapter 2 can be found in Burgschweiger et al. [BGS05]. In their work, the operative planning problem of water supply networks has been modeled as a pure NLP without integer variables. After that, a NLP solver has been used which usually ensures only locally optimal solutions. Very similar to this work, Blaszczyk et al. [BMA14] have modeled the optimal operation problem using NLP as well and solved it with the barrier method by extending solver Ipopt [Ipo].

Bragalli et al. [BDLLT06; BDLLT08; BDLLT12] published several related work to water supply networks. They have an MINLP model to make the choice of a diameter for each pipe, while other design properties are considered to be fixed, e.g., the network topology and pipe lengths. This kind of water network design problem is another big class of water supply network problems to compare with optimal operation problem. Their model contains discrete variables selecting from a set of commercially-available diameters. Water flows and pressures must respect the hydraulic constraints, and the total cost which only depends on the selected diameters should be minimized. In their work, they have tried several approaches to solve the MINLP, e.g., directly with BONMIN or solving MIP approach using piecewise-linear approximations.

In addition, gas networks, which also contain pumps (or compressors) and valves and should consider pressure, are very similar to the water supply networks. Martin et al. [MMM06] developed an MINLP model to optimize the flow of the gas and to use the compressors cost-efficiently such that all demands of the gas network are satisfied. But instead of solving the MINLP directly, techniques for a piecewise-linear approximation (see e.g., [BF76; LW01]) of the nonlinearities have been used to the model resulting in a large MIP. The remaining work is then to solve the MIP. The same approach has also be extended by [KLLMMOOR12] to solve operation of water supply networks. Morsi [Mor13] has compared the two problems in his PhD thesis in 2013.

For the most recent works, Perrière et al. [PJNN14] used an integer linear programming (ILP) tool for the MINLP. Another work is [FT15] where a second-order cone relaxation for the original MINLP model was proposed and solved to demonstrate the effectiveness in computing the optimal pump schedules and water flows.

Until now, we discussed mainly work with a focus on combinatorial aspects with a final goal of global optimization. There is also research focusing on numerical mathematics and continuous optimization by research group Martin and research group Lang [DGKLMM11; DKL15; KLB10; DKL10; GKLLMM11]. In these works differential equations for simulation purpose were underlying and should to be solved. In addition, sensitivity information for gradient-based optimization tools were provided. As a member of the research group Lang, Kolb [Kol11] finished his PhD thesis in 2011.

Indeed, MINLP is the most effective approach for water supply network problems recently. Recalling all previous work mentioned in this section, on the one hand, some of them used approximation already by modeling such that a subproblem, usually NLP. This reduces the solution quality due to the less precise model. ON the other hand, piecewise-linear approximations reduced the MINLP to be a MIP. First of all, the MIP with much more binary variables is not always easier solvable than the orgiginal MINLP. In addition, depending on the approximation parameter, very similar approximated MIPs could have very different optimal solutions. Other approaches such as sequential quadratic programming (SQP) [DGKLMM11] which extended a continuous treatment of binary control variables may handle the MINLP as well, but only ensures local optima.

Due to the significant improvement to mixed integer nonlinear programming solvers in recent years, larger and larger problems can be solved to optimality or near-optimality. We have then a chance of trying to develop an MINLP model as precise as possible, i.e., fewer relaxation and approximation comparing with the previous works. In this thesis, we try to solve it directly with our available MINLP solver and to get a global optimum if possible. To compare with previous work, we model in a more precise way and solve the modeled MINLP directly. This will definitely increase the solution quality for final operation. In addition, the approach with global optimality with gap information always verifies the solution quality numerically as well.

1.4 Introduction to mixed integer nonlinear programming

In this section we introduce definition, complexity, applications and solvers to mixed integer nonlinear programming (MINLP).

1.4.1 Definition

Definition 1.1 (Mixed Integer Nonlinear Program)
An optimization problem of the form

$$
\begin{aligned}
\text{min} \quad & d^T x & \\
\text{s.t.} \quad & g_i(x) \le 0 & i \in I \\
& L_j \le x_j \le U_j & j \in J \\
& x_k \in \mathbb{Z} & k \in J'
\end{aligned}
\tag{1.1}
$$

is a *Mixed Integer Nonlinear Program (MINLP)*, where I is the index set of constraints with $|I| = m$, J is the index set of all variables with $|J| = n$, $J' \subseteq J$ is the index set of integer variables, $d \in \mathbb{R}^n$, $g_i : \mathbb{R}^n \mapsto \mathbb{R}$, for all $i \in I$, and $L \in (\mathbb{R} \cup \{-\infty\})^n, U \in (\mathbb{R} \cup \{\infty\})^n$ are lower and upper bounds on the variables.

We call this MINLP in standard form. Note that an MINLP can have a nonlinear objective function, but it can easily be transferred to the standard form by introducing additional constraint(s) and variable(s).

If all of the constraint functions g_i are convex, the problem is known as a convex MINLP, otherwise it is known as a nonconvex MINLP. When all of the g_i are affine, we have a mixed integer linear program (MIP). Since MIP-solving is $\mathcal{NP}$-hard [Sch03, Volume A, Chapter 5] and MIP is a special case of MINLP, solving MINLP is at least $\mathcal{NP}$-hard. When $J' = \emptyset$, i.e., there are no integer variables, the problem becomes a nonlinear program (NLP). And if all of the g_i are affine and $J' = \emptyset$, we have a linear program (LP), which has been shown to be solvable in polynomial time by the ellipsoid method (see Khachian [Kha79]) and by interior point methods (see e.g., Karmarkar [Kar84]). Thus, both the integrality as well as the nonconvexity of g_i increase the complexity of solving an MINLP.

However, MINLP can not be solved exactly in general. Since neither the input nor the output of a Turing machine can be an irrational number, the solution of the simple constraint $x^2 = 2, x \geq 0$, which is an irrational number $\sqrt{2}$, cannot be computed or recognized by the Turing machine. On the other hand, no decimal presentation of a value of x can be verified to be an exact solution.

Plenty of optimization problems can be modeled as mixed integer programming (MIP) problems. However, for some applications, in particular in the field of some physical engineering systems, e.g., for gas network problems and for the operative planning of water supply networks, linear constraints cannot model the physical behavior accurately enough. In these cases we should model them as mixed integer nonlinear programming problems. MINLP has a wide range of applications, such as computational biology, computational chemistry, engineering design, etc. A survey of applications of MINLP can be found in Grossmann and Kravanja [GK97].

1.4.2 Solvers and algorithms

This overview is based on the presentation given in Bussieck and Vigerske [BV10]. One of the earliest commercial softwares package that could solve MINLP problems was SCICONIC in the mid 1970's (see also Forrest and Tomlin [FT07]). Instead of handling nonlinearities directly, Special-Ordered-Set constraints [BF76] were used to represent low dimensional nonlinear constraints by a piecewise linear approximation and thus allowed to use mixed integer programming (MIP) to get solutions to an approximation of the MINLP. In the mid 1980's Grossmann and Kocis developed DICOPT, a general purpose algorithm for convex MINLP based on the outer approximation method [DG86]. After that, a number of academic and commercial codes for convex MINLP were released. To solve nonconvex MINLPs to global optimality, the first general purpose solvers were ALPHABB [AM95], BARON [TS02a; Sah96], and GLOP [SP99], all based on convexification techniques for nonconvex constraints.

As presented in [BV10], algorithms for solving MINLPs are often built by combining algorithms from Linear Programming, Integer Programming, and Nonlinear Programming, e.g., branch-and-bound, outer-approximation, local search, and global optimization. Most of the solvers implement one (or several) of the following three algorithmic ideas [BV10]:

- Branch-and-bound solvers that use NLP relaxations. These solvers are e.g., BONMIN [BBC-CGLLLMSW08] (in B-BB mode), MINLP_BB [Ley01] and SBB [Oai]. For all these solvers, the NLP relaxation is obtained by relaxing the integrality conditions, thus it may be a nonconvex NLP. Since the NLP solver used to solve the NLP relaxation usually ensures only locally optimal solutions, these solvers work only as heuristics in case of a nonconvex MINLP.

- (Sequential) MIP-based solvers that replace the nonlinear functions by a linear relaxation. In an outer-approximation algorithm [DG86; FL94], a relaxation is obtained by using (sub)gradient-based linearizations of $g_i(x)$ at solution points of NLP subproblems. The resulting MIP relaxation is then solved by an MIP solver. Solvers in this class are e.g., BONMIN (in B-OA mode) and DICOPT [GVVRK02; Oai]. Since (sub)gradient-based linearizations yield an outer-approximation only for convex MINLPs, these solvers are only applicable for convex MINLPs. In contrast to outer-approximation based algorithms, an extended cutting plane algorithm solves a sequence of MIP relaxations which encapsulate optimal solutions and are improved by using cutting planes. This algorithm is implemented by the solver AlphaECP [WP95; WP02].

- Branch-and-cut solvers that integrate the linearization of $g_i(x)$ into a branch-and-cut process. In this process, an LP relaxation is successively solved, new linearizations are generated to improve the relaxation, and integrality constraints are enforced by branching on integer variables. Solvers which use gradient-based linearizations are e.g., AOA [RB09], BONMIN (in B-QG mode) and FilMINT [ALL06]. Gradient-based linearizations ensure global optimality only for convex MINLPs.

  For nonconvex MINLPs, convexification techniques can be used to compute linear underestimators of nonconvex functions. However, the additional convexification step may require branching on continuous variables in nonconvex terms (also called spatial branching). Such a branch-and-cut algorithm is implemented by e.g., BARON [TS02a; TS04], Couenne [BLLMW09] and SCIP [Ach07; BHV12; Vig12; BG12; Gle15].

1.5 Outline of the thesis

This thesis is structured as follows.

Chapter 2 gives a nonconvex MINLP model of the water supply network planning problem. In Chapter 3 we give a description of the static model which is the static version of the dynamic model introduced in Chapter 2. After that computational experiments show that the real-world instances can be solved to global optimality. However, computational results at the end of Chapter 4 and Chapter 5 show us that the instances with whole dynamic model are still very hard to solve.

In Chapter 4 we present theory and algorithms to simplify the derived MINLP after detecting passive sub-network. The simplified MINLP is easier to solve.

After that, in Chapter 5 we investigate first the characteristics of the convex hull of graphs of polynomial functions over a polytope. Based on the theoretical proofs, we develop algorithms to find tighter convex relaxations for the nonconvex polynomial objective functions in the MINLP model. The tighter convex relaxations improve the dual bound significantly and also helps solver to find better solution.

Finally in Chapter 6, we summarize our results and contributions and give a short outline of possible future research.

Chapter 2

Modeling Optimal Operation of Water Supply Networks by Mixed Integer Nonlinear Programming

In this chapter we present an MINLP model of our water supply network planning problem in Section 2.1. In Section 2.2, we introduce some simplification techniques as preprocessing steps for the original model without changing the optimality.

2.1 Optimization model

The network abstraction of our model and the notation of variables are based on [BGS05] and [Wal03]. The basic notation used in the model is given in Table 2.1.

2.1.1 Optimization horizon and network topology

Since our model is a time-expanded network which covers physical and technical network behavior, we consider a planning period of length T (typically one day, i.e., 24 hours) in discrete time, $t = 1, 2, \ldots, T$ with start status $t = 0$. We refer to the subinterval $(t-1, t)$ as period t which has length Δt. In our model, typically we set $\Delta t = 1$ hour, so there are 24 periods.

Our network model is based on a digraph $G = (\mathcal{N}, \mathcal{A})$, where the nodes represent junctions $\mathcal{J}$, reservoirs $\mathcal{R}$ and tanks $\mathcal{T}$, i.e.,

$$\mathcal{N} = \mathcal{J} \cup \mathcal{R} \cup \mathcal{T},$$

and the arcs represent pipe segments $\mathcal{S}$, pump stations $\mathcal{F}$ (facilities including pumps and equipment for pumping fluids) and gate valves $\mathcal{V}$, i.e.,

$$\mathcal{A} = \mathcal{S} \cup \mathcal{F} \cup \mathcal{V}.$$

Furthermore, $\mathcal{P}$ is the set of all pumps where every $p \in \mathcal{P}$ is contained in exactly one pump station.

Table 2.1: Notation for the optimization model.

Symbol	Explanation	Value	Unit
Q	Volumetric flow rate in arcs		m^3/s
Q^r	Volumetric flow rate out of reservoirs		m^3/s
h	Pressure potential at nodes (head)		m
h^l	Water fill level in tanks		m
Δh	Pressure increase at pumps, decrease at valves, pipes		m
x	Pump status	$\{0,1\}$	
y	Flow direction	$\{0,1\}$	
z	Variables that denote if the head is real or imaginary	$\{0,1\}$	
D	Demand flow rate at junctions		m^3/s
H^0	Geodetic elevation at nodes		m
V	Daily capacity at reservoirs		m^3
L	Pipe length		m
d	Pipe diameter (bore)		m
k	Pipe roughness		m
A	Pipe cross-sectional area		m^2
f	Pipe friction coefficient		$-$
λ	Pipe hydraulic loss coefficient		s^2/m^5
$\Delta H^{\max}$	Maximal possible pressure increase of pumps		m
c	Constant for characteristic curve of pumps		s^2/m^5
η	Efficiency of pumps		
ρ	Water density	1000	kg/m^3
g	Gravity constant	9.81	m/s^2
Δt	Length of a time period	3600	s
C	Total daily operating cost		Euro
C_a	Total daily operating cost at pump a		Euro
K_p	Cost of pump switch for pump p		Euro
κ_t	Price for electric energy at pump during period t		Euro$/J$
ω_j	Price of water in reservoir j		Euro$/m^3$

We denote arcs as $a \in \mathcal{A}$ or as ij, where i, j are the tail and head with $i, j \in \mathcal{N}$. For every $j \in \mathcal{N}$, let $\delta^+(j)$ be the set of arcs that have head j and $\delta^-(j)$ be the set of arcs that have tail j, i.e.,

$$\delta^+(j) := \{ji \in \mathcal{A} \mid i \in \mathcal{N}\},$$

and

$$\delta^-(j) := \{ij \in \mathcal{A} \mid i \in \mathcal{N}\}.$$

For an arc ij, a flow from i to j is positive and a flow from j to i is negative. Some arcs (e.g., pumps) may not permit negative flow.

2.1.2 Pressure

Since water is approximately not compressible, pressure p in unit Pa can be expressed as

$$p = \rho g \Delta h,$$

where ρ and g are constants and Δh is the height of water above the point of measurement or the elevation difference between the two points within the water column. To simplify our model, we can measure pressure only by the *head* h, which is the sum of the elevation difference Δh corresponding to the hydrostatic or hydraulic pressure and the geodetic elevation H^0:

$$h = \Delta h + H^0.$$

The geodetic elevation is the height above a fixed reference point; here, the mean sea level.

If some water flows from a reservoir without any external force, the *head* of this reservoir corresponds to its geodetic elevation. Water can only flow through a junction if the *head* is no less than the geodetic elevation of the junction. More details are given in Section 2.1.3 and Section 2.1.4.

2.1.3 Constraints

Junction. There are two kinds of junctions: junctions with demand (type 1) and junctions without demand (type 2). Recall $\mathcal{J}$ which is the set of all junctions, let $\mathcal{J}_1$ be the set of all junctions with demand and $\mathcal{J}_2$ be the set of all junctions without demand. Junctions with demand are actually the consumers. Junctions without demand are included in the network to connect two arcs, e.g. a junction between a pump and a pipe. For junctions, the consumption demands D_{jt} is a nonnegative constant (has value 0 for type 2) and has to be balanced,

$$\sum_{a \in \delta^-(j)} Q_{at} - \sum_{a \in \delta^+(j)} Q_{at} - D_{jt} = 0, \tag{2.1}$$

for all $j \in \mathcal{J}, t \in \{1, \ldots, T\}$.

Moreover, the *head* at every junction $j \in \mathcal{J}$ in every time period t has a lower bound. There are some complicated cases for junctions in the network, we discuss them in a separate section intentionally, namely in Section 2.1.4.

Reservoir. Every reservoir $j \in \mathcal{R}$ has a limited daily capacity V_j, from where the pure water flows out with flow rate Q^r:

$$\Delta t \sum_{t=1}^{T} Q^r_{jt} \leq V_j, \tag{2.2}$$

for all $j \in \mathcal{R}$. The outflow of every reservoir $j \in \mathcal{R}$ at every time period t should be balanced:

$$\sum_{a \in \delta^+(j)} Q_{at} - Q^r_{jt} = 0. \tag{2.3}$$

We assume that reservoirs have a constant pressure value H_j^0,

$$h_{jt} - H_j^0 = 0, \tag{2.4}$$

for all $j \in \mathcal{R}, t \in \{1, \dots, T\}$.

Tank. In our model all tanks are cylinders whose cross sections are invariant with area A_j. Flow balance equations at tank $j \in \mathcal{T}$ involve the tank inflow, which depends on $h_{j,t-1}$ and $h_{j,t}$,

$$\underbrace{\Delta t \left(\sum_{a \in \delta^-(j)} Q_{at} - \sum_{a \in \delta^+(j)} Q_{at} \right)}_{\text{total volume of inflow or outflow}} - \underbrace{A_j(h_{j,t} - h_{j,t-1})}_{\text{water volume difference}} = 0, \tag{2.5}$$

for all $j \in \mathcal{T}, t \in \{1, \dots, T\}$. For every tank $j \in \mathcal{T}$ in time $t \in \{1, \dots, T\}$ with actual water fill level $h_{jt}^l \in [F_j^{\min}, F_j^{\max}]$ and geodetic elevation H_j^0, the *head* h_{jt} should fulfill:

$$h_{jt}^l + H_j^0 = h_{jt} \tag{2.6}$$

Pipe. In every pipe $a = ij \in \mathcal{S}$, the hydraulic friction causes a pressure loss,

$$h_{jt} - h_{it} + \Delta h_{at} = 0,$$

for all $a \in \mathcal{S}, t \in \{1, \dots, T\}$.

The law of Darcy-Weisbach which has been expressed in [BGS05] and in [Wal03], presents the pressure loss in water pipes

$$\Delta h_{at} = \lambda_a Q_{at} |Q_{at}| = \lambda_a \text{sgn}(Q_{at}) Q_{at}^2, \tag{2.7}$$

where λ_a is the hydraulic loss coefficient which depends on the length L_a, the diameter d_a, and the friction coefficient f_a for every pipe $a \in \mathcal{S}$:

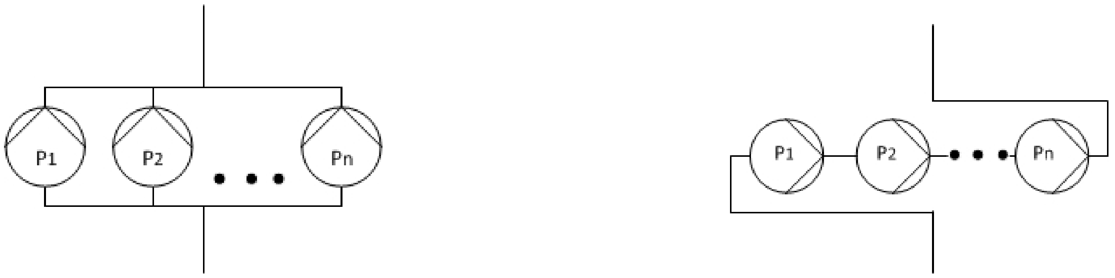
$$\lambda_a = \frac{8 L_a}{\pi^2 g d_a^5} f_a.$$

The friction coefficient f_a which has highly nonlinear dependency on the flow rate Q_a is taken into account by simulation software, see, e.g., EPANET [Epa], but appears to be too detailed for an optimization model.

We use the law of Prandtl-Kármán

$$f_a = \left(2 \log_{10} \frac{\varepsilon_a}{3.71 d_a} \right)^2,$$

which eliminates the dependency on Q_a by assuming large Reynolds number and is a good approximation for hydraulically rough pipes. It tends to underestimate the induced flow for small pressure differences, hence yielding conservative solutions. The roughness parameter ε_a only depends on the inner pipe surface. Since f_a is flow-independent, it follows λ_a is constant for every pump $a \in \mathcal{S}$. For more details on mathematical modeling of the physics of pressure loss, see, e.g., [BGS04].



(a) Connected in parallel

(b) Connected in series

Figure 2.1: Example of pump stations

Pump station. In our model pump stations may contain one single pump or several pumps. Pumps in the same pump station can be connected in parallel (Figure 2.1a) or in series (Figure 2.1b). Serially connected pumps increase the pressure one after another, and parallelly connected pumps increase the total flow rate and increase the pressure by an amount determined by the weakest pump of the group.

In our model we only consider pump stations which connect pumps in parallel. If a pump station has all pumps inactive, it functions as a completely closed valve: no flow through it and no pressure restriction. A pump station $a = ij \in \mathcal{F}$ that has more than one pump active in time $t \in \{1, \ldots, T\}$ increases the pressure by some nonnegative amount Δh_{at},

$$h_{jt} - h_{it} - \Delta h_{at} = 0. \tag{2.8}$$

For a given pump, the flow rate depends on the differential pressure or head developed by the pump. Such pumps have a curve of pump flow rate versus pump head, called characteristic curve, usually provided by the vendor. Let Δh_{pt} be the head of pump $p \in \mathcal{P}$ in time t and Q_{pt} be the flow rate. The characteristic curve [Epa] for pumps with fixed speed can be approximated by

$$\Delta h_{pt} = \Delta H_p^{\max} - c_p Q_{pt}^2, \tag{2.9}$$

where $\Delta H_p^{\max}$ and c_p are two positive constants derived from the characteristic curve. Note that $\Delta H^{\max}$ is the maximal possible pressure increase the pump can produce.

Let $Q_p^{\min}$ and $Q_p^{\max}$ be the lower and upper bound of the flow rate of pump $p \in \mathcal{P}$ during operation. If $Q_p^{\min} > 0$, the variable Q_{pt} is semi-continuous, i.e., we have $Q_{pt} \in \{0\} \cup [Q_p^{\min}, Q_p^{\max}]$. The main task of our operative planning problem is to decide the activity status and further the flow rate of all pumps $p \in \mathcal{F}$ during time period $t \in \{1, \ldots, T\}$. For every pump we define a binary variable $x_{it} \in \{0, 1\}$, where $x_{pt} = 1$ if and only if pump p is active in time period t. The flow rate Q_{pt} of pump $p \in \mathcal{P}$ during time period $t \in \{1, \ldots, T\}$ fulfills

$$x_{pt} Q_p^{\min} \leq Q_{pt} \leq x_{pt} Q_p^{\max}. \tag{2.10}$$

The flow rate in pump station $a \in \mathcal{F}$ in time t is the sum of the flow rate of every pump $p \in \mathcal{P}_a$:

$$Q_{at} = \sum_{p \in \mathcal{F}_a} Q_{pt}, \tag{2.11}$$

where $\mathcal{P}_a \subset \mathcal{P}$ is the set of all pumps contained in pump station a.

If pump p in pump station a is active, it should produce the same head Δh_{at} as the head between the two sides of a, but if it is inactive, it works just like a valve and its pressure differential Δh_{pt} does not have to be equal to Δh_{at}. For this purpose and in order to model them with linear constraints, we use a big-M formulation.

Remark 2.1
For clarity of presentation, we use the same constant M in all big-M constraints of our model. In our computations we choose M for each constraint individually as small as possible, depending on the bounds of the variables involved.

For every pump p in pump station a and every time t, we have

$$(x_{pt} - 1)M \leq \Delta h_{pt} - \Delta h_{at} \leq (1 - x_{pt})M. \tag{2.12}$$

If $x_{pt} = 1$, i.e., pump p is active in time t, then $\Delta h_{at} = \Delta h_{pt}$, i.e., the pump generates the same pressure increase as needed by the pump station. Otherwise if $x_{pt} = 0$, the constraint (2.12) will be fulfilled by all means.

For technical reasons we should have as few pump switches as possible. Pump p in time t has been switched if and only if

$$|x_{pt} - x_{p,t-1}| = 1.$$

Fortunately, one part of our objective is to minimize the number of pump switches, hence we can model the pump switch with two inequalities and an auxiliary variable Δx_{pt}:

$$\Delta x_{pt} \geq \pm(x_{pt} - x_{p,t-1}). \tag{2.13}$$

Apparently, since Δx_{pt} is not constrained by other constraints and has positive coefficient in the objective of a minimizing problem, an optimal solution fulfills

$$\Delta x_{pt} = |x_{pt} - x_{p,t-1}|.$$

For a given pump p, the efficiency η_{pt} in time t depends on the flow rate Q_{pt}. Similar to the characteristic curve, every pump has an efficiency curve of efficiency versus flow rate provided by the vendor. The efficiency curve can be approximated with three segments (see example in Figure 2.2):

$$\eta_{pt} = \begin{cases} a_{p1}Q_{pt} + b_{p1} & 0 \leq Q_{pt} \leq Q_{p1}, \\ a_{p2}Q_{pt} + b_{p2} & Q_{p1} < Q_{pt} \leq Q_{p2}, \\ a_{p3}Q_{pt} + b_{p3} & Q_{p2} < Q_{pt} \leq Q_{p3} \end{cases}$$

where $a_{p1}, a_{p2}, a_{p3}, b_{p1}, b_{p2}, b_{p3}, Q_{p1}, Q_{p2}, Q_{p3}$ are given coefficients for pump p.

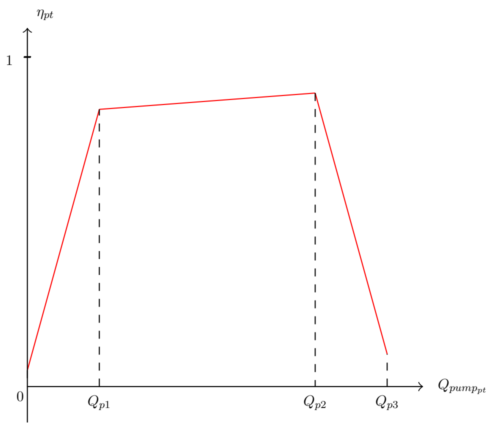


Figure 2.2: An example of how the pump efficiency depends on the flow rate

Valve. The pressure in valve $a = ij \in \mathcal{V}$ is decreased by some controlled amount Δh_{at},

$$h_{jt} - h_{it} + \Delta h_{at} = 0, \tag{2.14}$$

for all $a = ij \in \mathcal{V}, t \in \{1, \ldots, T\}$, and the sign condition

$$\Delta h_{at} Q_{at} \geq 0, \tag{2.15}$$

for all $a = ij \in \mathcal{V}, t \in \{1, \ldots, T\}$ guarantees the consistency of the pressure decrease.

However, the algorithm in this thesis used to solve MINLP performs better if the given MINLP has as few nonlinearities as possible. For this purpose, we replace the nonlinear constraints (2.15) by including a binary variable $y_{at} \in \{0, 1\}$ to determine the direction of the flow in valve a and time t. The flow rate Q_{at} is not negative if $y_{at} = 1$ and not positive if $y_{at} = 0$. Note that in case of $Q_{at} = 0$, the both values for y_{at} are allowed. Let $[\Delta h_a^{\min}, \Delta h_a^{\max}]$ be the domain of Δh_{at} and $[Q_a^{\min}, Q_a^{\max}]$ be the domain of Q_{at} with $Q_a^{\min} < 0, Q_a^{\max} > 0$ (otherwise we need no direction variable any more). Then (2.15) can be replaced by

$$\Delta h_a^{\min}(1 - y_{at}) \leq h_{it} - h_{jt} \leq \Delta h_a^{\max} y_{at}, \tag{2.16a}$$

$$Q_a^{\min}(1 - y_{at}) \leq Q_{at} \quad \leq Q_a^{\max} y_{at}. \tag{2.16b}$$

In (2.16) we have $\Delta h_{at} = h_{it} - h_{jt} \geq 0, Q_{at} \geq 0$ if $y_{at} = 1$ and $\Delta h_{at} = h_{it} - h_{jt} \leq 0, Q_{at} \leq 0$ if $y_{at} = 0$. Hence (2.15) is equivalent to (2.14) and (2.16). In addition, in case of $Q_{at} = 0$, the valve is closed completely and the head difference $h_{it} - h_{jt}$ is not constrained any more.

In our model valves have two types of functionalities:

- Gate valves.

  For every valve the constraint (2.15) must be fulfilled. Assume in a solution we have $Q_{at} = 0$ but $\Delta h_{at} \neq 0$. In this case, we close the valve completely, and at the same time the pressure difference is not controlled any more. It works like a gate to block the flow.

- Pressure decrease valves.

  Again, the consistency constraint (2.15) must be fulfilled. Assume in a solution we have Δh_{at} and Q_{at} are both positive or both negative, i.e., water flows through the valve with some pressure loss. In this case, we close the valve partially in order to decrease the pressure with $|\Delta h_{at}|$.

Remark 2.2
For those constraints in the following discussion which present the network in a single time period, we omit time horizon to simplify the discussion. Note that they have the same form and are appropriate for every time period.

2.1.4 Real and imaginary flow

In Section 2.1.2 we mentioned that we can measure pressure by *head* H on every node in the network, and in Section 2.1.3 there are *head* variables h_{it} defined for the *head* at node i and in time t, where node i can be a junction with demand (consumer), a junction with no demand, a reservoir, or a tank. Note that pressure really exists in a node only if there is water flowing through it (if it is a junction) or if there is water stored in it (if it is a reservoir or a tank). Since in our model tanks[1] and reservoirs are never empty, the pressure in tank and in reservoir always exists.

As explained above, different pressure levels at the ends of a pipe induce nonzero flow according to the law of Darcy-Weisbach as given by equation (2.7). However, this only holds if water is indeed present at the high-pressure node. With active elements like closed valves or inactive pumps, pipes have no water flow. In this case, strict enforcement of (2.7) leads to a physically unsound model.

As an example, consider the subnetwork shown in Figure 2.3 taken from the real-world instance in Figure 3.3 introduced in Section 3.3.1. An elevated tank t_1 is connected to the network via valve k_1. Pipe s_3 leads downwards, i.e., $H^0_{j_2} > H^0_{j_1}$. Suppose now valve k_1 is closed. By flow balance, $Q_{s_3} = 0$, and for (2.7) to hold we need $h_{j_1} = h_{j_2}$, i.e., the head at j_1 must lie strictly above its geodetic height. In reality, however, the subnetwork functions as if s_3, j_2, k_1, and t_1 were not present, hence $h_{j_1} = H^0_{j_1}$ might be a valid state.

We call head levels at nodes without flowing water and the flow that would be induced by these head levels according to the law of Darcy-Weisbach *imaginary* as opposed to *real.* In the above example, the incorrect assumption was to enforce equation (2.7) although the head at j_2 is imaginary in solutions with closed valve k_1.

Remark 2.3
So far we have not seen this distinction being made in the literature. Although it may be that depending on the structure of the network all head levels can be validly assumed to be real, we believe this to be a potential source for harmful modeling gaps. Note that this distinction is

[1] Tanks could be empty in the reality, but in our model, we never let tanks be empty in order to safeguard against the underestimation of consumers' demands.

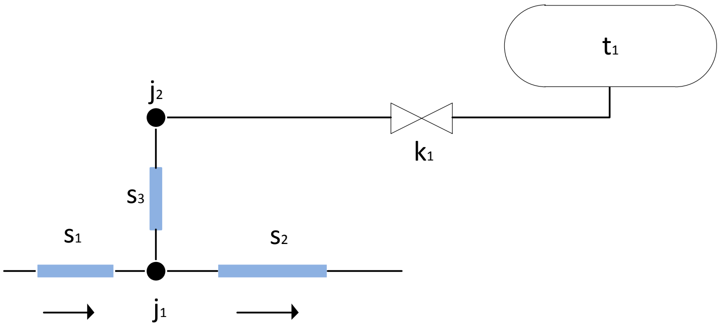


Figure 2.3: Subnetwork with imaginary flow for closed valve k_1.

equally necessary for the full-scale operative planning problem and can be made by the same constraints proposed here.

We introduce a binary variable z_j at each node $j \in \mathcal{N}$ to distinguish between real ($z_j = 1$) and imaginary ($z_j = 0$) heads. The variable z_j is forced to be 1 if the head is strictly greater than its geodetic height,

$$h_j \leqslant H_j^0 + M z_j, \tag{2.17}$$

or if flow passes through j, i.e.,

$$-M z_j \leqslant Q_a \leqslant M z_j \tag{2.18}$$

for all $a \in \delta(j)$. Water supply networks are usually operated such that water sources such as reservoirs and tanks are never completely empty and may be assumed as real, so we set $z_j = 1$ for all $j \in \mathcal{R} \cup \mathcal{T}$.

Furthermore, we need to model how water is propagated along pipes. If a pipe ij is horizontal then water is present at i if and only if it is present at j, i.e.,

$$z_i = z_j \tag{2.19}$$

for all $ij \in \mathcal{S}$ with $H_i^0 = H_j^0$. For pipes with nonzero slope two implications hold. First, if the geodetically higher node, node i, say, is real, so is the lower node j,

$$z_i \leqslant z_j. \tag{2.20}$$

Second, if the lower node j is real and contains water with higher pressure than H_i^0, then also i must be real,

$$h_j \leqslant H_i^0 + M z_i. \tag{2.21}$$

Finally, we enforce equation (2.7), the law of Darcy-Weisbach, between (and only between) real nodes by

$$\Delta h_a = \lambda_a \mathrm{sgn}(Q_a) Q_a^2$$

and

$$M(z_i + z_j - 2) \leqslant h_i - h_j - \Delta h_a \leqslant M(2 - z_i - z_j) \tag{2.22}$$

for all pipes $a = ij \in \mathcal{S}$.

Remark 2.4

Note that both in reality and in our model a node may be real in spite of zero flow through the node: $z_j = 1$ and $Q_a = 0$ for all $a \in \delta(j)$. As an example, imagine an additional, closed valve at node j_1 in Figure 2.3, while valve k_1 is open. Then pipe s_3 would be completely filled with water from the tank, hence nodes j_1 and j_2 would be real. At the same time, the water column in the pipe yields pressure $h_{j_1} = h_{j_2}$ and so the law of Darcy-Weisbach is satisfied by zero flow, $Q_{s_3} = 0$.

2.1.5 Objective

Because of constraints (2.9), Δh_{pt} in objective (2.24) can be replaced by $\Delta H_p^{\max} - c_p Q_{pt}^2$. The energy cost per time unit (second) of pump p in time t is presented [Epa] as

$$C_{pt} = \frac{\kappa_t \rho g \Delta h_{pt} Q_{pt}}{\eta_{pt}} = \frac{\kappa_t \rho g \left(\Delta H_p^{\max} Q_{pt} - c_p Q_{pt}^3\right)}{\eta_{pt}} \tag{2.23}$$

Since $\kappa_t, \rho, g, \Delta H_p^{\max}$ and c_p are constant and the value of η_{pt} depends only on Q_{pt}, C_{pt} is a univariate function of Q_{pt}.

The goal is to minimize the sum of the energy cost from pumps and the cost of water from reservoirs as well as the "cost", i.e., the number of pump switches. The objective is therefore

$$K = \underbrace{\Delta t \sum_{t=1}^{T} \sum_{j \in \mathcal{R}} \omega_j Q_{jt}^r}_{\text{cost of water}} + \underbrace{\Delta t \sum_{t=1}^{T} \sum_{p \in \mathcal{P}} \frac{\kappa_t \rho g \Delta h_{pt} Q_{pt}}{\eta_{pt}}}_{\text{cost of pumps}} + \underbrace{\sum_{t=1}^{T} \sum_{p \in \mathcal{P}} k_p \Delta x_{pt}}_{\text{“cost” of pump switches}} \tag{2.24}$$

Consider η_{pt} again, which is the denominator of C_{pt}. The efficiency curve of efficiency versus flow rate can be approximated with three segments, see the example in Figure 2.2 again. If the flow rate is very small, e.g., $< Q_{p1}$ or very big, e.g., $> Q_{p2}$, the efficiency is then very low which implies that the energy cost C_{pt} is very high. For this purpose, we restrict the flow rate further into the interval $[Q_{p1}, Q_{p2}]$. After that, we need only one segment for the efficiency curve. Since the segment is almost parallel to the x-axis, we can handle the efficiency as a constant. For that, we should make slight change to $Q_p^{\min}$ and $Q_p^{\max}$ which are the lower and upper bound of the flow rate and used in (2.10). Note that the first part and the third are linear, which are convex and concave, with the concave function $\Delta H_p^{\max} Q_{pt} - c_p Q_{pt}^3$, we have a concave objective.

Although the objective contains several parts, we need not to handle the optimization problem as a multiobjective optimization problem, since there is real cost in every part of the objective. More details about multiobjective optimization can be found in [Ehr05].

An equivalent objective to (2.24) is

$$K = \underbrace{\Delta t \sum_{t=1}^{T} \sum_{j \in \mathcal{R}} \omega_j Q_{jt}^r}_{\text{cost of water}} + \underbrace{\Delta t \sum_{t=1}^{T} \sum_{p \in \mathcal{P}} C_{pt}}_{\text{cost of pumps}} + \underbrace{\sum_{t=1}^{T} \sum_{p \in \mathcal{P}} k_p \Delta x_{pt}}_{\text{“cost” of pump switches}} \ , \tag{2.25}$$

which is linear.

2.1.6 Summary of the model

Let $\mathrm{TP} = \{1, \ldots, T\}$ be the set of all time periods. The complete nonconvex MINLP now reads

$$\left.\begin{array}{lll}
\min & \Delta t \sum_{t=1}^{T} \sum_{j \in \mathcal{R}} \omega_j Q_{jt}^r + \Delta t \sum_{t=1}^{T} \sum_{p \in \mathcal{F}} C_{pt} + \sum_{t=1}^{T} \sum_{p \in \mathcal{F}} k_p \Delta x_{pt} & \\
\text{s.t.} & ((2.1) - (2.13)), ((2.16) - (2.22)), (2.23), & \\
& x_{pt} \in \{0,1\}, Q_{pt} \in [0, Q_p^{\max}], \Delta h_{pt} \in [0, \Delta H_p^{\max}] & \text{for all } p \in \mathcal{P}, t \in \mathrm{TP}, \\
& y_{vt} \in \{0,1\}, Q_{vt} \in [Q_v^{\min}, Q_v^{\max}] & \text{for all } v \in \mathcal{V}, t \in \mathrm{TP}, \\
& z_{jt} \in \{0,1\}, h_{jt} \in [H_j^0, H_j^{\max}] & \text{for all } j \in \mathcal{N}, t \in \mathrm{TP}, \\
& z_{jt} = 1 & \text{for all } j \in \mathcal{R} \cup \mathcal{T}, t \in \mathrm{TP}, \\
& Q_{at} \in [Q_a^{\min}, Q_a^{\max}] & \text{for all } a \in \mathcal{S} \cup \mathcal{F}, t \in \mathrm{TP}, \\
& h_{jt}^l \in [F_j^{\min}, F_j^{\max}] & \text{for all } j \in \mathcal{T}, t \in \mathrm{TP}, \\
& Q_{jt}^r \in [Q_j^{\min}, Q_j^{\max}] & \text{for all } j \in \mathcal{R}, t \in \mathrm{TP}.
\end{array}\right\} \tag{2.26}$$

The original objective (2.24) is nonlinear, which is also nonconvex. The further nonlinear constraints are quadratic constraints (2.7) as well as constraints (2.9). Except for these, all other constraints are linear. For integrality conditions there are binary variables x_{pt} for pump status, binary variables y_{at} for flow direction in valves as well as binary variables z_{it} which denote if the head is real or imaginary.

2.2 Reformulation and presolving

This section outlines a set of straightforward problem-specific presolving steps that help to reduce both size and difficulty of given instances of type (2.26). The reductions explained in the following are exact in the sense that a feasible solution is cut off only if another essentially identical solution remains.

2.2.1 Contracting subsequent pipes

Suppose a zero demand junction j is incident with two pipes, one entering ij, and one leaving jk. Flow balance enforces $Q_{ij} = Q_{jk} =: \tilde{Q}$ and if nonzero flow passes through j the Darcy-Weisbach equations read $h_i - h_j = \lambda_{ij}\mathrm{sgn}(\tilde{Q})\tilde{Q}^2$ and $h_j - h_k = \lambda_{jk}\mathrm{sgn}(\tilde{Q})\tilde{Q}^2$. These two constraints are equivalent to

$$h_i - h_k = (\lambda_{ij} + \lambda_{jk})\mathrm{sgn}(\tilde{Q})\tilde{Q}^2$$

and

$$h_j = \frac{\lambda_{jk}h_i + \lambda_{ij}h_k}{\lambda_{jk} + \lambda_{ij}}.$$

We want to exploit this to replace pipes ij and jk by a new, aggregated pipe ik with friction coefficient $\lambda_{ij} + \lambda_{jk}$ and consequently remove junction j from the network.

In case nonzero flow $\tilde{Q} \neq 0$ is guaranteed to pass through the pipe, we only need to ensure satisfiability of $h_j \geqslant H_j^0$ by

$$\frac{\lambda_{jk}h_i + \lambda_{ij}h_k}{\lambda_{jk} + \lambda_{ij}} \geqslant H_j^0. \tag{2.27}$$

To account for $\tilde{Q} = 0$, however, we need to keep variable z_j in the model, since it may be zero even if $z_i = z_k = 1$. (As an example consider the case that junction j is located much higher than i and k and can hence block flow even if water is available at i and k.)

Darcy-Weisbach holds if and only if all three nodes i, j, and k have real head, i.e., constraint (2.22) becomes

$$M(z_i + z_j + z_k - 3) \leqslant h_i - h_k - \Delta h_{ik} \leqslant M(3 - z_i - z_j - z_k). \tag{2.22a}$$

Constraints ((2.17)–(2.21)) involving z_j remain unchanged. To ensure (2.27) if j is real, we add constraint

$$\frac{\lambda_{jk}h_i + \lambda_{ij}h_k}{\lambda_{jk} + \lambda_{ij}} \geqslant H_j^0 - M(1 - z_j). \tag{2.28}$$

The cases of two pipes entering or leaving a zero demand junction work analogously. Pipe sequences with several inner nodes $ij_1, j_1j_2, \dots, j_Nk$ can be treated similarly – for each inner node we only need to add its z variable to ((2.22)a) and include constraint (2.28).

Note that these presolving steps do not just yield a smaller problem, but most importantly removes nonlinear equations of type (2.7).

J1
J2
J3
S1
S2
J1
J3
S12

(a) Before contracting

(b) After contracting

Figure 2.4: Contracting pipe sequences

2.2.2 Breaking symmetry in pump stations

As we mentioned in Section 2.1, in our model we have only pump stations whose pumps are connected in parallel, see example in Figure 2.1a. For pump p_i and pump p_j in the same pump station, we say that p_i can replace p_j if p_i and p_j have the same characteristic curve and the domain of the flow rate of p_j is contained in the domain of the flow rate of p_i. Further if p_i and p_j have the same domain of flow rate, they can replace each other, in this case we say p_i and p_j are equivalent. For an arbitrary pump configuration Ψ_1 which sets p_i inactive and sets p_j active in time t, we can easily find another pump configuration Ψ_2 which makes small changes to Ψ_1: set p_i active and set p_j inactive in time t. These two pump configurations are apparently equivalent except for the pump status of p_i and p_j. In this case, we call Ψ_1 and Ψ_2 symmetric pump configurations.

Let $a \in \mathcal{F}$ be a pump station which contains pumps $\mathcal{F}_a = \{p_1, p_2, \dots, p_n\}, n \geq 2, n \in \mathbb{N}$. Let $\{i_1, i_2, \dots, i_s\} \subseteq \{1, 2, \dots, n\}, 2 \leq s \leq n$ be an index set of pumps (we call it also priority set) in $\mathcal{F}_a$ and for each $j, 1 \leq j \leq s-1$, pump p_{i_j} can replace pump $p_{i_{j+1}}$. In order to avoid symmetric pump configurations, we set priority to these pumps: for every $k \in \mathbb{N}$ with $1 \leq k \leq s-1$, pump p_{i_k} can only be set to be active if at the same time all p_{i_q} have been set to be active, $q \in \mathbb{N}$, $k < q \leq s$. Let $x_{p_{i_j}t} \in \{0, 1\}$ be the pump status of pump p_{i_j} in time t where 1 means active and 0 means inactive, after setting the priority of pump status we have

$$x_{p_{i_1}t} \leq x_{p_{i_2}t} \leq \dots \leq x_{p_{i_s}t}. \tag{2.29}$$

In Model (2.26), one part of the objective (2.24) is to minimize the number of pump switches. Since changing the pump status locally can change the number of pump switches, we are concerned if the priority set of pumps can violate the optimality of Model (2.26).

Theorem 2.5
Let Ψ_1 be a pump configuration corresponding to an optimal solution s_1 of Model (2.26). There exists a pump configuration Ψ_2 which also corresponds to an optimal solution s_2 and fulfills all priority requirements (2.29) of pumps.

Proof. We assume that Ψ_1 does not fulfill all priority requirements of pumps, otherwise we need only to set $\Psi_2 := \Psi_1$.

Let t be time with $t \in \{1, \dots, T\}$. For every given priority set $\{i_1, i_2, \dots, i_s\}$ of pumps to a pump station, we count how many of them in Ψ_1 have been set to be inactive in time t. Let r

be the number of inactive pumps in this priority set in time t, we then set the first r pumps in this priority set for Ψ_2 to be inactive and the rest of them to be active. For pumps which appear in none of the priority sets, we copy the status of Ψ_1 into Ψ_2. Thus Ψ_2 is a pump configuration corresponding to a feasible solution s_2 which may contain different number of pump switches to s_1.

$$\begin{array}{rl} t: & \overbrace{00\cdots0}^{r_1}\overbrace{11\cdots1}^{|r_1-r_2|}11\cdots1 \\ t+1: & \underbrace{00\cdots0\,00\cdots0}_{r_2}11\cdots1 \end{array}$$

Now we only need to prove that the number of pump switches in Ψ_2 is not more than the number of pump switches in Ψ_1. For every time t and every priority set $\{i_1, i_2, \ldots, i_s\}$, let r_1 be the number of inactive pumps in this priority set in time t and r_2 be the number of inactive pumps in this priority set in time $t+1$. Then the number of pump switches of all pumps in this priority set is at least $|r_1 - r_2|$. Since Ψ_2 has set the first r_1 pumps inactive in time t and set the first r_2 pumps inactive in time $t+1$, the number of pump switches is exactly $|r_1 - r_2|$. Hence s_2 has not more pump switches than s_1, the optimality of s_1 follows that s_2 is optimal.□

As a result, solving Model (2.26) with priority sets does not affect its optimality and the priority set (2.29) reduces the search space for feasible choices of active pumps significantly from 2^s to $s+1$. Solving MINLP needs branching on integer variables in general, the priority sets can decrease the number of branching.

2.2.3 Contracting pipe-valve-sequences

Suppose a pipe $ij \in \mathcal{S}$ and a valve $jk \in \mathcal{V}$ are connected by a zero demand junction j. Flow balance enforces $Q_{ij} = Q_{jk} =: \tilde{Q}$. Figure 2.5 shows the feasible values of pressure loss $h_k - h_i$ versus $\tilde{Q}$. While the Darcy-Weisbach equation forces the pressure loss along the pipe onto the dashed line, the valve allows for larger pressure loss in absolute value. The feasible region is hence a union of two convex sets, the dotted area for backward flow and the shaded area for forward flow.

This can be exploited replacing pipe ij and valve jk by a new arc $a = ik$ and relaxing valve constraints (2.16a) and (2.16b) and pipe constraints (2.7) and (2.22) to

$$M(y_a - 1) \leqslant Q_a \leqslant M y_a \tag{2.16c}$$

for flow direction as before,

$$\Delta h_a \geqslant \lambda_{ij} Q_a^2 \tag{2.7a}$$

for the minimum pressure loss, and

$$M(z_i + y_a - 2) \leqslant h_i - h_k - \Delta h_a \tag{2.22b}$$

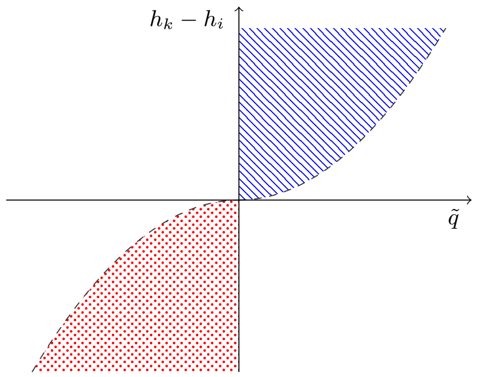


Figure 2.5: Feasible values of pressure loss versus flow through a pipe-valve-sequence $ij \in \mathcal{S}$, $jk \in \mathcal{V}$.

and

$$h_i - h_k + \Delta h_a \leqslant M(1 - z_k + y_a) \tag{2.22c}$$

for the relaxed Darcy-Weisbach equation.

This reduction replaces the nonconvex, nonconcave constraint (2.7) by a convex quadratic constraint ((2.7)a). Again, other combinations of arc directions work analogously.

2.2.4 Fixing and propagating z variables

At junctions with nonzero demand, flow balance requires nonzero flow on at least one incident arc. Trivially, (2.18) implies that the head is real:

$$j \in \mathcal{J}, D_j > 0 \Longrightarrow z_j = 1.$$

Using these fixings and the water sources known to be real, some of the constraints ((2.19)—(2.21)) may then become redundant or can be used to fix further z variables to one.

2.2.5 Handling special cases for junctions without demand

In Section 2.1.4 we mentioned that for a given zero demand junction i we can only require that its head H_{it} is no less than its geodetic height $\bar{H}_i$ if the head is real, i.e., there is nonzero flow through it. There is an example shown in Figure 2.3 that the constraint $H_{it} \geq \bar{H}_i$ may change the optimal value if H_{it} is imaginary. For this, our solution is including a binary variable which denotes if the head is real or imaginary and then using constraints ((2.17)-(2.22)). Now we want to find some special cases for the given junction j and prove in these cases binary variable z_{jt} as well as constraints ((2.17)-(2.22)) for junction j from Model (2.26) are redundant:

- Assume that we have a pipe a whose two connected nodes are j_1 and j_2. Node j_1 is a zero demand junction and besides a, j_1 is only connected with another arc that is not a pipe. In addition, the geodetic height of j_1 is not higher than the geodetic height of j_2, namely $H^0_{j_1} < H^0_{j_2}$. Otherwise, in case of $H^0_{j_1} \geq H^0_{j_2}$, (2.20) is applicable.

  Consider the case $z_{j_2} = 1$, $H^0_{j_1} < H^0_{j_2}$ follows then flow has enough pressure to reach node j_1. Then we have $z_{j_1} = 1$. On the other hand, if $z_{j_2} = 0$, flow balance follows that $Q_a = 0$. In this case we have $z_{j_1} = 0$. Then we can add the valid constraint

  $$z_{j_1} = z_{j_2}$$

  to Model (2.26).

- Very similar to the case above, assume that we have a pipe a whose two nodes are j_1 and j_2. j_2 is a zero demand junction and besides a j_2 is only connected with another arc but not a pipe. We assume further that the geodetic height of j_1 is much higher than the geodetic height of j_2, namely $H^0_{j_1} \gg H^0_{j_2}$. "Much higher" means here $H^0_{j_1} - H^0_{j_2} > (Q^{\max}_{at})^2$. Assume for some reason we know that the flow direction in a must be from j_1 to j_2. If the head in j_1 is real with $z_{j_1} = 1$, then flow is reachable to j_2 which means $z_{j_2} = 1$. On the other hand, if $z_{j_2} = 0$, flow balance follows that $Q_a = 0$. In this case we have $z_{j_1} = 0$. Then we can add the valid constraint

  $$z_{j_1} = z_{j_2}$$

  to Model (2.26).

The simplification techniques above either focus on the reduction of network size which leads to an MINLP with fewer constraints and variables, or focus on changing the structure of integrality and nonlinearity of the Model. Both do not change the feasibility and the optimality of the original model and make the MINLP easier to solve.

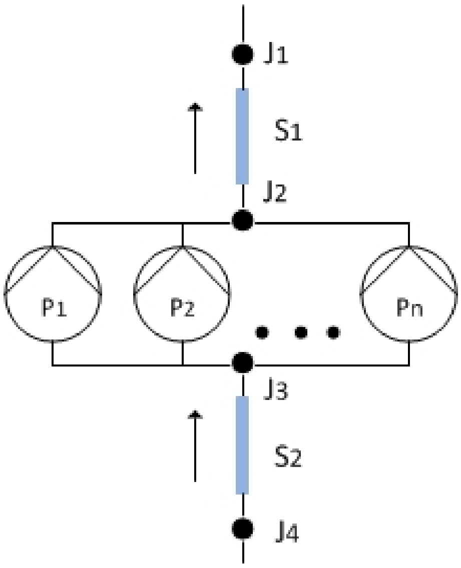


Figure 2.6: Special cases for junctions without demand

Chapter 3

Solving Optimal Operation of Water Supply Networks in a Fixed Point of Time

The full-scale optimal operation task typically covers the time span of one day with hourly demand forecasts for each consumer and comprises the decision when and which pumps are switched on and off, when and which tanks are filled for storage (typically at low demand), when they are deflated again (typically at peak demand), and where water is obtained.

In this chapter, we focus on solving a stationary version of the planning problem to ε-global optimality: Given fixed starting levels of tanks and constant demands of consumers, compute a pump configuration and a feasible flow through the network such as to minimize the variable operational cost incurred by purchase of energy and water. Although this alone does not address the full operative planning task, it arises naturally, e.g., as one-period subproblem in a time-discretized formulation. In heuristic or decomposition-based solution approaches these subproblems may have to be solved iteratively. Not least, the ability to compute proven optimal solutions to these stationary models can help in evaluating and improving heuristic solution techniques.

The content of Chapter 3 has already be published in [GHHV12]. This chapter is organized as follows. Section 3.1 gives the static model of the optimal operation problem as a *mixed-integer nonlinear program* (MINLP) which is the static version of the dynamic model introduced in Section 2.1. Section 3.2 explains how this can be solved to globally proven optimality gaps. Section 3.3 presents results of computational experiments conducted on real-world instances provided by our industry partner Siemens AG, Corporate Technology, Modeling, Simulation & Optimization.[1] Finally, Section 3.4 contains concluding remarks.

3.1 The model in a fixed point of time

The goal of this chapter is to optimize the operation of a water supply network at a fixed point in time. Given filling levels of tanks and demands of consumers we wish to compute a feasible flow through the network such as to minimize the variable operational cost incurred by purchase of energy and water. In the following, we model this problem as a nonconvex *mixed-integer nonlinear program* (MINLP).

[1] `http://www.ct.siemens.com/`

Table 3.1: Variables of the optimization model.

variable	interpretation
h_j	pressure potential (head) at node $j \in \mathcal{N}$ $[m]$
Δh_a	pressure increase/decrease at pump or pipe $a \in \mathcal{P} \cup \mathcal{S}$ $[m]$
Q_a	volumetric flow rate in arc $a \in \mathcal{A}$ $[m^3/s]$
Q_w	volumetric outflow rate from water source $w \in \mathcal{W}$ $[m^3/s]$
x_p	binary indicator whether pump $p \in \mathcal{P}$ is switched on
y_v	binary indicator for direction of valve $v \in \mathcal{V}$
z_j	binary indicator whether head at node $j \in \mathcal{N}$ is real

The main difference between the static model and the dynamic model introduced in Section 2.1 is the behavior of tanks. In the dynamic case we consider that the filling level of tanks changes by time and flow. However, for the model in a fixed point of time, we assume that the filling level will not change, then it works as a reservoir. For that we define then the set of water sources $\mathcal{W} = \mathcal{T} \cup \mathcal{R}$ which contains all tanks and reservoirs. For every water source $w \in \mathcal{W}$, Q_w denotes the outflow from w, then for the flow balance we have

$$\sum_{a \in \delta^+(w)} Q_a - \sum_{a \in \delta^-(w)} Q_a - Q_w = 0. \tag{3.1}$$

Without consideration of time t, Table 3.1 summarizes the variables used in our static optimization model. The complete nonconvex MINLP now reads

$$\left.\begin{array}{lll} \min & \sum\limits_{j \in \mathcal{R}} \omega_j Q_j^r + \sum\limits_{p \in \mathcal{F}} C_p & \\ \text{s.t.} & (2.1), (2.4), ((2.6)-(2.13)), ((2.16)-(2.22)), (2.23), (3.1), & \\ & x_p \in \{0,1\}, Q_p \in [0, Q_p^{\max}], \Delta h_p \in [0, \Delta H_p^{\max}] & \text{for all } p \in \mathcal{P}, \\ & y_v \in \{0,1\}, Q_v \in [Q_v^{\min}, Q_v^{\max}] & \text{for all } v \in \mathcal{V}, \\ & z_j \in \{0,1\}, h_j \in [H_j^0, H_j^{\max}] & \text{for all } j \in \mathcal{N}, \\ & Q_a \in [Q_a^{\min}, Q_a^{\max}] & \text{for all } a \in \mathcal{S} \cup \mathcal{F}, \\ & z_w = 1, Q_w^r \in [Q_w^{\min}, Q_j^{\max}] & \text{for all } w \in \mathcal{W}. \end{array}\right\} \tag{3.2}$$

As we discussed before, it features two types of nonlinearities, the Darcy-Weisbach equation (2.7) along each pipe and the energy consumption of pumps in (2.23), both of which are nonconvex. Together with the discrete states encoded in the binary variables this yields a highly nonconvex solution space.

3.2 Global solution approach

The problem formulation given in the previous section is a nonconvex MINLP. Its combination of discrete and continuous nonconvexities – binary decision variables for pump status, valve direction, and imaginary flow plus nonconvex nonlinear terms (2.7) and (2.23) – results in a challenging optimization problem. In the following we describe how well-known algorithmic techniques can be applied to solve them to ε-global optimality.

3.2.1 Branch-and-bound

A common methodology to handle nonconvex optimization problems is *branch-and-bound* [LD60], where the problem is successively divided into smaller subproblems until the individual subproblems are sufficiently easy to solve. Additionally, bounding is used to detect early whether improving solutions can be found in a subproblem and avoid enumerating suboptimal parts of the feasible region. Thereby, bounds on the optimal objective function value are computed from a computationally tractable relaxation of the current subproblem.

For nonconvex MINLPs, typically an efficiently solvable convex (linear or nonlinear) relaxation is used for *bounding*, obtained by dropping integrality conditions and replacing nonconvex nonlinear functions by convex estimators [TS04]. *Branching* (problem division) is done with respect to either discrete variables that take a fractional value in the relaxation's solution or variables that appear in violated nonconvex constraints. The purpose of the latter is that a reduction of a variable's domain yields tighter convex estimators, which in turn may allow to cut off the infeasible solution from the relaxation.

Solvers that implement branch-and-bound algorithms for general MINLPs are BARON [TS04], Couenne [BLLMW09], LINDO API [LS09], and SCIP [Ach09; Vig12] etc. By default, all of them employ a linear relaxation.

We used the solver SCIP, a framework for solving constraint integer programs by a branch-and-bound algorithm. Arguably, from the solvers listed above, it provides the strongest support for solving mixed-integer programs (MIPs), which is necessary to address the combinatorial aspect of our optimization problem. Its state-of-the-art MIP features include cutting plane separators, primal heuristics, domain propagation algorithms, and support for conflict analysis [Ach07; Ach09]. SCIP has been extended to handle also nonlinear constraints since a few years [BHV12; Vig12].

3.2.2 Outer approximation

For the nonlinear functions $Q_a \mapsto \lambda_a \mathrm{sgn}(Q_a) Q_a^2$ from constraint (2.7) and $Q_p \mapsto \Delta H_p^{\max} Q_p - c_p Q_p^3$ from constraint (2.23), SCIP generates a linear outer approximation along their convex and concave envelopes. If the relaxation's solution violates nonlinear constraints, the outer approximation is tightened by branching on the flow variables Q_a and Q_p. For $Q_a \mapsto \lambda_a \mathrm{sgn}(Q_a) Q_a^2$, this is illustrated in Figure 3.1. For further details, we refer to [Vig12].

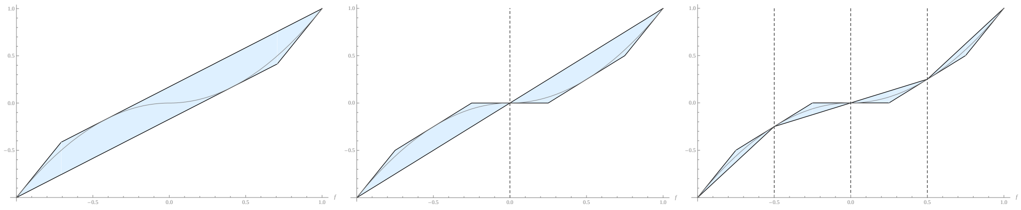


Figure 3.1: Linear outer approximation of the nonlinear function $Q_a \mapsto \lambda_a \mathrm{sgn}(Q_a) Q_a^2$ and the effect of branching on Q_a.

To improve performance, SCIP uses the constraints to propagate a reduction in one variable's domain to other variables. For example, if the bounds on variable Δh_a in constraint $\Delta h_a = \lambda_a \mathrm{sgn}(Q_a) Q_a^2$ are reduced to $[\underline{\Delta h_a}, \overline{\Delta h_a}]$, the bounds of Q_a can be tightened to

$$\left[\mathrm{sgn}(\underline{\Delta h_a})\sqrt{|\underline{\Delta h_a}|/\lambda_a}, \mathrm{sgn}(\overline{\Delta h_a})\sqrt{|\overline{\Delta h_a}|/\lambda_a}\right],$$

which allows for a tighter linear outer approximation. Similarly, tighter bounds for Δh_a may be deduced from domain reductions for Q_a.

3.2.3 Primal solutions

Although in theory, it suffices to collect feasible solutions of the relaxation at leaves of the branch-and-bound tree, in practice, it is highly beneficial to apply heuristic procedures interleaved with the global search. Finding good solutions early in the search allows the user to stop the solution early if he is already satisfied with the achieved solution quality. Algorithmically, better primal bounds allow the branch-and-bound tree to be pruned earlier and can hence improve solver performance.

SCIP uses several primal heuristics to find feasible solutions early in the search. First, SCIPs default MIP primal heuristics [Ber06; Ber14] are applied to find a point that is feasible for the linear relaxation plus the integrality requirements, but may violate some of the nonlinear constraints. Subsequently, the binary variables (x, y, z) are fixed to their value in this solution and the resulting nonlinear program (NLP) is solved to local optimality using Ipopt [WB06]. If the NLP is feasible, any solution is also feasible for the original MINLP.

Second, SCIP employs various large neighbourhood search heuristics extended from MIP to MINLP [Ber06; BHPV11; Ber14] or specifically designed for MINLP [BG10; BG12]. These heuristics use the relaxation solution or previously found feasible solutions to construct a hopefully easier sub-MINLP by restricting the search space, e.g., via variable fixings. The reduced problem is then partially solved by a separate SCIP instance.

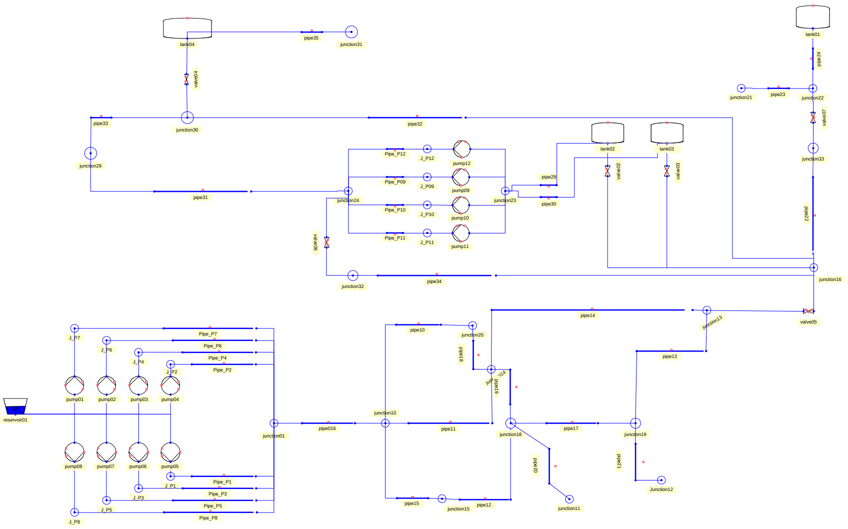


Figure 3.2: Schematic diagram of water supply network n25p22a18 with 25 nodes (1 reservoir, 4 tanks, 20 junctions), 4 consumers, 22 pipes, 12 pumps, and 6 valves.

3.3 Computational experiments

3.3.1 Instances

This section presents the results of our computational experiments on two networks provided by our industry partner Siemens AG. Figure 3.2 shows a small water supply network n25p22a18 on 25 nodes (1 reservoir, 4 tanks, 20 junctions), 4 consumers, 22 pipes, 12 pumps, and 6 valves. The second network n88p64a64 on 88 nodes (15 reservoirs, 11 tanks, 62 junctions), 22 consumers, 64 pipes, 55 pumps, and 9 valves is depicted in Figure 3.3. Each network comes with hourly demand forecast for one day. Both are real-world water supply networks.

3.3.2 Experimental setup

The goal of our experiments was to investigate whether and how fast the stationary version of the operative planning problem in form of the MINLP model (3.2) can be solved to ε-global optimality and to evaluate the computational impact of the presolving reductions described in Section 2.2.

Exemplarily, we selected the demand forecasts for 0-1 am (low demand), 6-7 am (first peak demand), 12-1 pm (medium demand), and 6-7 pm (second peak demand). The results for these scenarios were representative for the other hours.

For the tank levels, we considered two scenarios. In the *medium tank level scenario*, we assume all tanks to be half-full; in this case, a large portion of the demand may be satisfied by emptying the tanks only, without significant pump activity. However, such a solution will

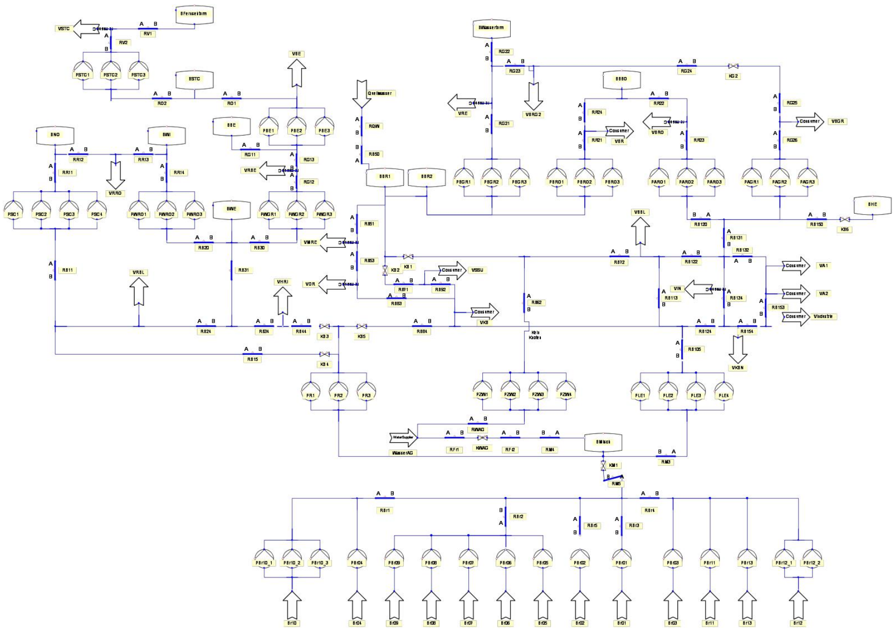


Figure 3.3: Schematic diagram of water supply network instance n88p64a64 with 88 nodes (15 reservoirs, 11 tanks, 62 junctions), 22 consumers, 64 pipes, 55 pumps, and 9 valves.

be very greedy and also the difficulty of the MINLPs may be reduced. Therefore, for a second test, we select the tanks that—if the first solution was implemented—would run empty first and set them to their minimum filling level, hence only allowing for inflow into these tanks; for network n25p22a18 we reset the first, for n88p64a64 we reset the first four tanks that would run empty to their minimum filling levels. We refer to this as *low tank level scenario.*

For our experiments we solely used academic software that is available in source code. We ran SCIP 2.1.1 [Sci] with SoPlex 1.6.0 [Sop] as LP solver, Ipopt 3.10.1 [Ipo] as NLP solver, CppAD 20110101.5 [Cpp] as expression interpreter for evaluating nonlinear functions, and Zimpl 3.2 [Zim] as modeling language. SCIP was run with default settings and a time limit of one hour. We conducted the experiments on an AMD Opteron 6174 with 2.2 GHz and 128 GB RAM. Note that the computations were executed before the result has been published in [GHHV12]. Since all MINLP instances have been solved within time limit, we do not repeat the computations again. All other computations in this thesis have been finished with the recent version of SCIP.

3.3.3 Computational results

First, we evaluate the impact of the problem-specific presolving steps described in Section 2.2. After these steps new MINLP instances are prepared. Table 3.2 shows how these help to reduce the size of the problems in number of variables “vars”, binary variables “bin”, number of

Table 3.2: Problem sizes without and with problem-specific presolving as described in Section 2.2.

network	without presolving				with presolving			
	vars	bin	cons	nlin	vars	bin	cons	nlin
n25p22a18	145	28	332	42	139	24	322	40
n88p64a64	561	99	1098	171	542	81	982	170

constraints “cons” and number of nonlinear constraints “nlin”. Note that the problem reductions apply to the structure of the network and are independent of demand forecast or tank levels. The numbers given are computed before applying SCIP’s presolving. Fixed variables and bound constraints are not counted. The largest reduction occurs in the number of binary variables, which are reduced by 14% and 18%, respectively. The number of nonlinear constraints is only slightly reduced.

Table 3.3 compares running times and number of branch-and-bound nodes explored by SCIP when solving to optimality with a tolerance of 10^{-6}. It can be seen that the scenarios for the smaller instance n25p22a18 can all be solved within one second and can only improve minimally when using presolving. The most difficult instances are the low tank level scenarios for the larger network n88p64a64. Here, both solution time and number of branch-and-bound nodes decrease drastically when applying presolving. Due to smaller branch-and-bound trees, the instances are solved faster by a factor between 3.8 and 89.5. The only slowdown occurs on “0-1 am med” and “6-7 am med” because SCIP’s primal heuristics do not find the optimal solution at the root node anymore. Nevertheless, these are solved within less than two seconds. All in all, the presolving steps presented in Section 2.2 proved highly beneficial in our experiments.

Finally, Table 3.4 presents our computational results for the presolved instances in more detail. From column “objval” listing the objective value of the optimal solution found, we can confirm the expectation that the low tank level scenarios always require more pumps being active, except for demand “6-7 pm” in n25p22a18, where the objective value remains at the same level. In all cases, the “low” scenarios take at least as long as the “med” scenarios. In particular for n88p64a64, this seems to explain why the “med” scenarios are computationally much easier: a solution with no active pumps is feasible and can be found and proven to be optimal very fast.

The last three columns analyze the solution progress in more detail, giving the time to find a first feasible solution, the time to achieve a proven primal-dual gap of 5%, and the time until an optimal solution is found. A gap of 5% is always reached within 2.4 seconds except for n88p64a64 “12-1 pm low”, where it takes 16.7 seconds. In almost all cases, the optimal solution is found at the very end of the solution process. For the instance n88p64a64 “6-7 pm low” with longest running time of 104 seconds, however, the situation is reversed: the optimal solution is found already after 1.5 seconds and SCIP spends the remaining time to prove its

Table 3.3: Running times and number of branch-and-bound nodes to optimal solution without and with presolving as described in Section 2.2.

	scenario		without presolving		with presolving	
	demands	tanks	time	nodes	time	nodes
n25p22a18	0-1 am	med	0.7s	247	0.4s	67
		low	0.9s	663	0.8s	85
	6-7 am	med	0.6s	219	0.4s	60
		low	1.0s	478	0.8s	77
	12-1 pm	med	0.5s	76	0.6s	76
		low	1.0s	239	0.9s	172
	6-7 pm	med	0.5s	54	0.5s	80
		low	0.4s	54	0.5s	80
n88p64a64	0-1 am	med	0.4s	1	1.1s	75
		low	11.2s	3518	1.1s	16
	6-7 am	med	0.6s	1	1.6s	181
		low	595.4s	334128	12.8s	5495
	12-1 pm	med	3.6s	1044	2.4s	430
		low	1941.4s	1195329	21.7s	6738
	6-7 pm	med	4.2s	1413	1.0s	85
		low	399.8s	236966	104.0s	64940

optimality.

3.4 Concluding remarks

This chapter has presented a small contribution to the task of optimal, i.e., energy- and cost-minimal, optimal operation of water supply networks. The research in this chapter has focused on a stationary version of this challenging optimization problem and aimed at ε-globally optimal solution techniques. The MINLP model used is detailed in the sense that it incorporates the nonlinear physical laws as well as the discrete decisions involved.

Through computational experiments on instances from industry, we demonstrated that the stationary models presented can be solved to global optimality within small running times using problem-specific presolving and a state-of-the-art MINLP solution algorithm. On one hand, this verified the correctness of instance data and the modeling; one the other hand, the research in this chapter shows us that we can also hope that we my reach the global optimality to the dynamic model (2.26).

Table 3.4: Detailed computational results for water supply networks n25p22a18 and n88p64a64 after presolving as described in Section 2.2.

	scenario		optimality			time		
	demands	tanks	objval	time	nodes	first sol	5% gap	best sol
n25p22a18	0-1 am	med	42.63	0.4s	67	0.2s	0.4s	0.4s
		low	64.55	0.8s	85	0.5s	0.6s	0.8s
	6-7 am	med	42.51	0.4s	60	0.2s	0.2s	0.4s
		low	62.82	0.8s	77	0.5s	0.7s	0.8s
	12-1 pm	med	60.54	0.6s	76	0.3s	0.6s	0.6s
		low	72.78	0.9s	172	0.8s	0.8s	0.9s
	6-7 pm	med	60.54	0.5s	80	0.1s	0.2s	0.5s
		low	60.54	0.5s	80	0.1s	0.2s	0.5s
n88p64a64	0-1 am	med	0	1.1s	75	1.1s	1.1s	1.1s
		low	4.45	1.1s	16	0.7s	1.1s	0.7s
	6-7 am	med	0	1.6s	181	1.6s	1.6s	1.6s
		low	118.76	12.8s	5495	0.6s	0.9s	12.8s
	12-1 pm	med	0	2.4s	430	2.4s	2.4s	2.4s
		low	86.58	21.7s	6738	12.2s	16.7s	21.7s
	6-7 pm	med	0	1.0s	85	1.0s	1.0s	1.0s
		low	51.24	104.0s	64940	0.8s	1.0s	1.5s

Chapter 4

Acceleration of Solving MINLPs by Symbolic Computation

This chapter is concerned with solving a subNLP of the MINLP model (2.26) which corresponds to a subnetwork of the entire network for the optimal operation of pressurized water supply networks. Subnetworks which contain only pipes and junctions yield a constrained nonlinear system (CNS, a nonlinear program without objective). We show how the CNS can be replaced by a single (or a few) nonlinear constraint(s) after variable elimination without changing the feasible region and how faster the MINLPs can be solved after the replacement.

4.1 Introduction and motivation

Recall all components contained in a general water supply network. They are pumps in pump stations, valves, pipes, junctions (some are related to customers), reservoirs and tanks. Getting an operation which yields available and reliable water supply is the same as finding a configuration for all pumps and values. From the point view of mathematical programming, a configuration corresponds to fixing those configuration variables with reasonable values. We call the other variables *passive variables* since we cannot set their values by configurations. These are the variables for pipes, junctions, reservoirs and tanks. Furthermore, reservoirs are usually connected to pumps directly, the outflow from them depends actually on the status of pumps which corresponds to configuration variables. In addition, the fill level of tanks varies with time. It follows that a subnetwork that contains only pipes and junctions can be regarded as a passive and static network. An example of such a subnetwork is shown in Figure 4.1. In this subnetwork, only two junctions are connected with the rest of the network: we call them *joint junctions*. Variables and constraints which are related to this subnetwork form a constrained nonlinear system (CNS) and a subset of the MINLP for the optimal operation problem.

During the survey of literature we found some research for the similar subnetworks, mainly focuses on uniqueness of solutions as well as algorithms. Birkhoff et al. [BD56] have already started to observe the uniqueness property in the 1950s. After that, Collins et al. [CCVHLKJL78] presented a pair of different optimization models both of which are equivalent to the CNS. Hence solving CNS is equivalent to solving convex optimization problems. Similarly, Maugis [Mau77] gave a reduction for this particular CNS to a strictly convex problem which always contains a

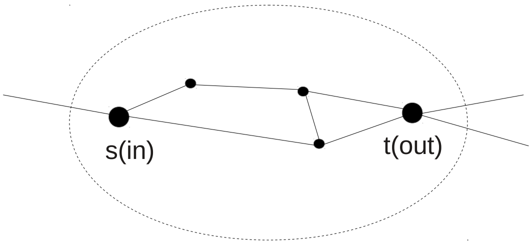


Figure 4.1: A semi-passive subnetwork

unique solution. Thus the property of unique solvablity has been shown to have a reasonable complexity. Similarly, Ríos-Mercado et al. [RMWSB02] have proved the unique solvablity property for gas networks. A most recent work about algorithmic results for general potential-based flows can be seen in [GPSSS17]. To compare with the literature above, we prove the unique solvablity property in a topological way.

Note that the variables in the CNS which also appear in the other constraints of the MINLP are only the variables for the joint junctions. The variables for other junctions and for pipes are passive variables, if these can be eliminated, i.e., there exists an equivalent CNS which contains all variables for the joint junctions while the original and the equivalent CNS both have the same feasible region for the variables of joint junctions. Thus the current CNS can then be replaced by the new CNS in the MINLP. If the new CNS has less variables or even less nonlinearities, the replacement reduces the dimension or the complexity of the original MINLP. In the following we show that we can always reduce the original MINLP by replacing CNSs which are related to such subnetworks by their equivalent CNSs that have some variables eliminated and contain therefore less constraints (nonlinearities). We call the new MINLP *reduced* MINLP.

Chapter 4 is organized as follows. Section 4.2 describes a CNS model related to these subnetworks which forms a subproblem of the model introduced in Section 2.1. After that, we prove that the CNS has exactly one unique solution by fixing the value of a selected variable. In addition, for every variable of the CNS, there exists a function which maps the selected variable to this variable. The proof is given in Section 4.2.

Section 4.3 employs symbolic computation to get these functions. In this section, we design first an algorithm to split the original CNS in several similar polynomial equation systems which are a precondition for symbolic computation. In the following, we eliminate all median variables by computing the Gröbner bases of the corresponding polynomial equation systems and get the function in symbolic form by computing the roots of univariate polynomials.

Section 4.4 presents results of computational experiments. Unfortunately, the exact functions found in Section 4.3 are so complicated that the MINLP solver SCIP cannot use them directly. However, polynomials with degree 2 can fit these exact univariate functions with quite acceptable errors. Instead of using totally exact CNSs to replace the original CNSs to get a reduced MINLP, we use approximated CNSs to get an approximated MINLP. Our computations have verified the correctness of approximated MINLPs and shown that the new MINLPs are much easier to solve.

4.2 Model

4.2.1 Network description and classification

The network we are observing in this chapter is a connected subnetwork of a water supply network. Let $G = (\mathcal{N}, \mathcal{A})$ be a given water supply network defined in Section 2.1.1. In this chapter we work essentially with subnetworks $G_s = (\mathcal{N}_s, \mathcal{A}_s) \subset G$ with the following properties:

- $\mathcal{N}_s \subset \mathcal{J} \subset \mathcal{N}$ contains only *junctions* j and $\mathcal{A}_s \subset \mathcal{S} \subset \mathcal{A}$ contains only *pipes* a,
- For all $i, j \in \mathcal{N}_s$, if $a = (i, j) \in \mathcal{A}$, then a is a pipe and $a \in \mathcal{A}_s$,
- There are exactly two junctions, say inflow note s and outflow note t, which are connected to the remaining graph,
- G_s is connected.

We call the graph G_s a semi-passive subnetwork. Later we will give the reason for the name.

After that, we can define the remaining graph $G' = (\mathcal{N}', \mathcal{A}')$ with $\mathcal{N}' = (\mathcal{N} \setminus \mathcal{N}_s) \cup \{s, t\}$ and $\mathcal{A}' = \mathcal{A} \setminus \mathcal{A}_s$.

An example of such a subnetwork is shown in Figure 4.1. In general, we regard the graph as an undirected graph since the flow direction in pipes is not determined. However, for formulating the constraints with mathematical programming we want to define a direction which does not prescribe the direction of flow through this element, but only indicates the meaning of a positive flow.

Recall the variables and constraints related to this subnetwork which have been introduced in the full MINLP model in Section 2.1.1. Each pipe a carries a signed flow Q_a and on each junction j, the pressure is measured by head h_j.

Remark 4.1
As a tradition for the discussion of the network flow problems, we use s to denote the source node and t for the sink node. As a result, we use t' to denote time in this chapter.

Our model is a time-expanded network. We consider a planning period of length T (typically one day, i.e., 24 hours) in discrete time, $t' = 1, 2, \ldots, T$ with start time $t' = 0$. In general, the variables $Q_{at'}$ and $h_{jt'}$ are time-dependent. In this chapter, we restrict attention only to these subnetworks. All related constraints are only related to one time period. Note that all constraints which are discussed in this chapter are time-independent. Thus we omit the time t' for all variables to simplify our discussion.

At each junction $j \in \mathcal{N}_s$ except of the two special nodes s and t, the flow balance equation

$$\sum_{a\in\delta^-(j)} Q_a - \sum_{a\in\delta^+(j)} Q_a - D_j = 0, \tag{2.1}$$

has already been defined in Section 2.1. Junctions with positive demand $D_j > 0$ correspond to consumers, all others satisfy $D_j = 0$.

In addition, the total inflow Q_s into s from the remaining graph can be defined as

$$Q_s = \sum_{a=is\in\mathcal{A}'} Q_a - \sum_{a=si\in\mathcal{A}'} Q_a.$$

Similarly, the total inflow Q_t into t from the remaining graph can be defined. With a slight modification of (2.1), we get the flow balance equations for s and t

$$\sum_{a=(i,j)\in\mathcal{A}} Q_a - \sum_{a=(j,i)\in\mathcal{A}} Q_a + Q_j = D_j \tag{4.1}$$

where $j = s$ or $j = t$ and D_j a constant. Adding flow balance equations (2.1) for all $j \in \mathcal{N}_s \setminus \{s,t\}$ and (4.1) for $j \in \{s,t\}$, it follows then

$$Q_s + Q_t = \sum_{j\in\mathcal{N}} D_j =: D \geq 0.$$

Since D is a nonnegative constant, either Q_s or Q_t should be nonnegative and one can decide the value of the other. Usually, we regard a nonzero Q_s or Q_t to be inflow if they are positive or outflow if they are negative.

To simplify our discussion, we call s the inflow node and t the outflow node without loss of generality. Note that the signs of Q_s and Q_t are not restricted from the name of s and t.

The flow of water through a pipe $a = (i,j)$ is a function of the pressure levels h_i and h_j at its ends. The pressure loss along the pipe is described by the law of *Darcy-Weisbach*

$$h_i - h_j = \lambda_a Q_{at} \left|Q_a\right| = \lambda_a \mathrm{sgn}(Q_a) Q_a^2, \tag{2.7}$$

which has been defined in Section 2.1.

In addition, the head at each node should be no less than its geodetic height H_j^0 if the head is real:

$$h_j \geqslant H_j^0. \tag{4.2}$$

Consider the constraints ((2.1), (2.7), (4.1)) related to a given semi-passive subnetwork G_s, the head variables h_j do not appear in ((2.1), (4.1)) and only appear pairwise in (2.7). For any solution of constraints ((2.1), (2.7), (4.1), (4.2)), increasing the head at every node by $h' > 0$ will construct a new solution since all increased h_j will fulfill the constraints ((2.7), (4.2)) as well.

Remark 4.2
To model the full water network we need additional constraints for the operation of components pumps, valves, reservoirs and tanks. In this chapter, we first restrict our attention to subnetworks that do not contain these components. Therefore the constraints above suffice to model the behavior of these subnetworks. For a full description of the overall subnetwork see e.g., [GHHV12; Hua11; BGS05].

In the following discussion we ignore the constraints (4.2) first and consider them again at the end of this section.

4.2.2 Unique solvability

In the following, we are concerned with solving the constrained nonlinear system CNS(G_s).

Definition 4.3 (CNS(G_s))
Given a semi-passive network G_s, we define CNS(G_s) as a constrained nonlinear system containing constraints ((2.1), (2.7), (4.1)) and set $h_s = 0$. They are summarized as

$$\left.\begin{aligned} \sum_{a\in\delta^-(j)} Q_a - \sum_{a\in\delta^+(j)} Q_a - D_j = 0 \quad & \text{for all } j \in \mathcal{N}_s \setminus \{s,t\}, \\ \sum_{a=(i,j)\in\mathcal{A}} Q_a - \sum_{a=(j,i)\in\mathcal{A}} Q_a + Q_j - D_j = 0 \quad & \text{for all } j \in \{s,t\}, \\ h_i - h_j = \lambda_a Q_{at}\,|Q_a| - \lambda_a \mathrm{sgn}(Q_a) Q_a^2 = 0 \quad & \text{for all } a=(i,j)\in\mathcal{A}_s, \\ h_s = 0 \quad & \end{aligned}\right\} \qquad (\text{CNS}(G_s))$$

As we discussed above, the head variables h_j appear pairwise in CNS(G_s). To eliminate solutions which have the same value of Q_a, we fix $h_s = 0$. It follows then

$$h_j - h_s = h_j$$

for every node $j \in \mathcal{N}_s$, which means that the value of h_j is the head difference between node j and node s.

Consider a semi-passive subnetwork in the entire water supply network again. To operate the water supply network means to find a feasible configuration of pumps and valves. However, the semi-passive subnetwork does not contain any pump and valve which we can control. From the natural point of view, if the network operates with $Q_s = Q_s^0$ for a constant $Q_s^0 \in \mathbb{R}$, there has to be a unique solution for CNS(G_s). We verify this mathematically with the following theorem.

Theorem 4.4 (Unique Solvability of CNS(G_s))
For a given semi-passive subnetwork G_s and any constant $Q_s^0 \in \mathbb{R}$, CNS(G_s) has a unique solution with $Q_s = Q_s^0$. Furthermore, for every node j there exists a continuous, decreasing or constant function

$$f_j : \mathbb{R} \to \mathbb{R} \text{ with } h_j = f_j(Q_s)$$

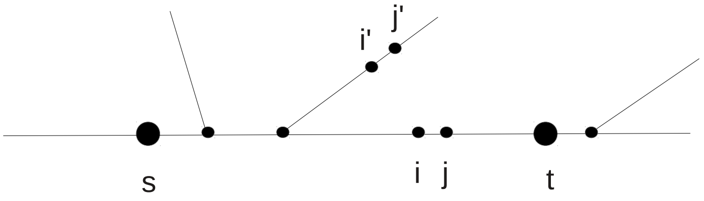


Figure 4.2: A semi-passive subnetwork in tree structure

which maps the inflow Q_s into node s from the remaining graph to the head h_j at node j; for every arc a there exists a continuous function

$$g_a : \mathbb{R} \to \mathbb{R} \textit{ with } Q_a = g_a(Q_s)$$

which maps the inflow Q_s into node s from the remaining graph to the flow Q_a through pipe a. The function g_a is either constant or monotonic. To be increasing or decreasing depends on the definition of the direction for positive flow.

Proof. Let G_s be the given semi-passive subnetwork with $m := |\mathcal{A}_s|, n := |\mathcal{N}_s|$. Since G_s is connected we have $m \geq n-1$ which is equivalent to $m - n + 1 \geq 0$. Define $x := m - n + 1$ then we have $x \in \mathbb{N}_0, x \geq 0$. Note that for connected graph, x denotes the number of cycles.

Now we want to prove that all semi-passive subnetworks with $m - n + 1 = x = 0, 1, 2, \ldots$ possess the properties with mathematical induction over x.

The first step is to prove the theorem is true for the case $x = 0$. Consider the subnetworks with $m-n+1 = x = 0$, i.e., $m = n-1$. In this case the graph is a tree, see e.g., Figure 4.2. For every arc $a \in \mathcal{A}_s$, removing a yields two disjoint connected graphs: the left graph $G_a^l = (\mathcal{N}_a^l, \mathcal{A}_a^l)$ with $s \in \mathcal{N}_a^l$ and the right graph $G_a^r = (\mathcal{N}_a^r, \mathcal{A}_a^r)$. For every arc a, we discuss first how the value of Q_a depends on Q_s. Since the graph has a tree structure, this is a unique path from s to t. For every arc a in this graph there are two cases:

- Arc a is not on the path from s to t, see e.g., arc $a = i'j'$ in Figure 4.2. Due to flow balance, the flow on arc a, i.e., from i' to j' has to be equal to the total demand of all nodes of G_a^r. It follows that
$$Q_a = \sum_{n \in \mathcal{N}_a^r} D_n =: D_a^r.$$
Flow Q_a is a constant since D_a^r is a constant.
- Arc a is an arc on the path from s to t, see e.g., arc $a = ij$ in Figure 4.2. The left graph G_a^l has inflow Q_s and total demand $D_a^l := \sum_{n \in \mathcal{N}_a^l} D_n$, the remaining flow from G_a^l which flows from i to j is then
$$Q_a = Q_s - D_a^l.$$

Note that we may define the flow direction from j to i to be the positive direction, then we would have

$$Q_a = -(Q_s - D^l_a).$$

Obviously, for both cases there exists a function g_a for every arc a which fulfills the corresponding properties.

Now we discuss how the value of h_j depends on Q_s. For every node $j \in \mathcal{N}$, there is a unique path from s to j since G_s is a tree. Let the path be $n_0, n_1, \ldots, n_p$ with $n_0 = s$, $n_p = j$ and $p \in \mathbb{N}$. The arcs on the path are $a_r = (n_{r-1}, n_r)$ for all $r \in \{1, \ldots, p\}$. Moreover, we can split the set $S := \{1, \ldots, p\}$ into two sets S_1 and S_2 so that $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \emptyset$ and for every $r \in S_1$ it holds $t \in G^l_{a_r}$ and for every $r \in S_2$ it holds $t \in G^r_{a_r}$. The function f_j can be represented as

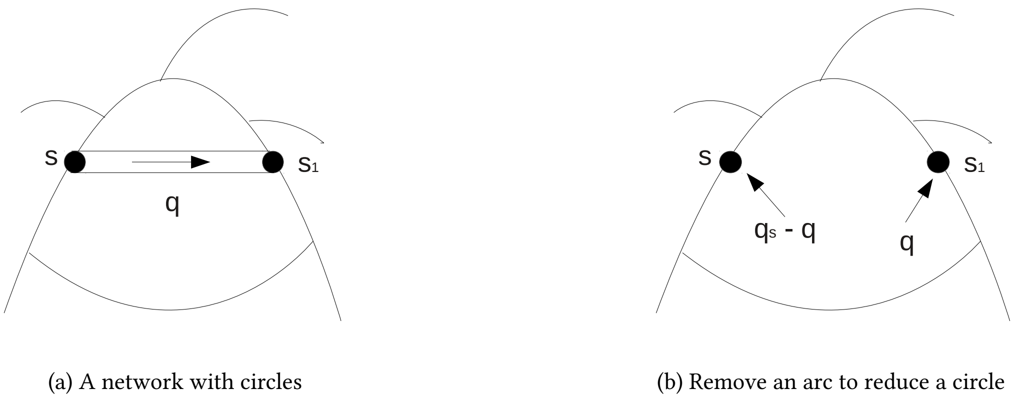
$$\begin{aligned} h_j &= h_j - 0 = -(h_s - h_j) \\ &= -\left((h_s - h_{n_1}) + (h_{n_1} - h_{n_2}) + \ldots + (h_{n_{p-1}} - h_j)\right) \\ &= -\sum_{r=1}^{p} \lambda_{a_r} \operatorname{sgn}(Q_{a_r}) Q^2_{a_r} \\ &= -(\sum_{r \in S_1} \underbrace{\lambda_{a_r} \operatorname{sgn}(D^r_{a_r})(D^r_{a_r})^2}_{\text{constant}} + \sum_{r \in S_2} \underbrace{\lambda_{a_r} \operatorname{sgn}(Q_s - D^l_{a_r})(Q_s - D^l_{a_r})^2}_{\text{increasing function of } Q_s}) \\ &=: f_j(Q_s). \end{aligned}$$

Note that f_j is a decreasing function if $S_2 \neq \emptyset$ or a constant function otherwise.

Setting $Q_s = Q^0_s$ yields the unique solution of CNS(G_s). Until now we proved that the theorem is true for $x = m - n + 1 = 0$, i.e., for all graphs with $m = n - 1$. Suppose that the theorem is true for all graphs with $x = k$, i.e., $m = n - 1 + k, k \in \mathbb{N}_0$. We need only to prove that the theorem is also true for all graphs with $x = k + 1$, i.e., $m = (n - 1 + k) + 1 = n + k$.

Let G_s be a semi-passive subnetwork of type $m = n + k$, then G_s contains at least one circle since connected networks are circle-free if and only if $m = n - 1$.

In general, s does not have to be contained in a cycle, see e.g., Figure 4.4. All neighboring arcs (s, s_r) to s are not contained in a circle, for $r = 1, \ldots, n_c$, n_c is a constant with $n_c \geq 1$. Consider all possible paths from s to t. All of these contain exactly one of the arcs (s, s_r) for $r = 1, \ldots, n_c$. Without loss of generality, (s, s_1) is contained in path(s) from s to t. Since every (s, s_r) is not contained in a cycle, the flow $Q_{(s,s_r)}$ can be calculated to be a fixed value as we have shown during proving the case of $x = 0$. For any $r \neq 1$, removing (s, s_r) leads to two subgraphs. The subgraphs which contains node s_r contains neither s nor t. All flow and head variables can be solved trivially. After we remove all arcs (s, s_r) with $r \neq 1$, node s is connected only to (s, s_1). Now we remove arc (s, s_1) and set s_1 to be the new inflow note with $Q_{s_1} = Q_s - Q_{(s,s_1)}$ so that we generate an equivalent new CNS problem. Note that we moved the inflow node from s to s_1. After doing the procedure above recursively, we will move inflow note to a note which is contained in a cycle.



(a) A network with circles

(b) Remove an arc to reduce a circle

Figure 4.3: Semi-passive subnetworks

Now we need only to discuss the case that s is contained in a cycle. See an example in Figure 4.3a, from s there is an arc $a = (s, s_1)$ contained in a circle.

For any given network G_s as shown in Figure 4.3a, we construct an auxiliary network G_s^o as shown in Figure 4.3b by

- removing arc (s, s_1),
- setting the demand of (the original) t for G_s in G_s^o to be $D - Q_s$ with total demand D,
- setting s for G_s as s for G_s^o with inflow $Q_s - q$ by introducing new variable q with $q \in \mathbb{R}$ and setting s_1 as t for G_s^o with inflow q.

Note that G_s^o is still connected since (s, s_1) is contained in a circle. For any given $Q_s \in \mathbb{R}$ (inflow of s in G), G_s^o is a semi-passive subnetwork of type $m = n + k - 1$ with inflow $Q_s - q$. With the induction hypothesis, for the head $h_{s_1}^o$ at s_1 in G_s^o there exists a function $f_{s_1}^o$ with $h_{s_1}^o = f_{s_1}^o(Q_s - q)$ which is a continuous, decreasing or constant function. With $a = (s, s_1)$ in G, let p_a be the pressure loss function of pipe a in G, then we have $h_{s_1} = -p_a(Q_a)$. For $q \in \mathbb{R}$ a given constant, let $Q_s - q$ be the inflow into s for G_s^o. The unique solution of CNS(G_s^o) is equivalent to a solution of CNS(G_s) if and only if

$$h_{s_1}^o = f_{s_1}^o(Q_s - q) = -p_a(q) = h_{s_1} \text{ and } Q_a = q.$$

For a given Q_s the value of q satisfies

$$F(q) := f_{s_1}^o(Q_s - q) + p_a(q) = 0.$$

Since $f_{s_1}^o$ is a continuous, constant or decreasing function, then for a fixed given Q_s, $f_{s_1}^o(Q_s - q)$ is then a continuous, constant or increasing function of q. Together with p_a which is a continuous,

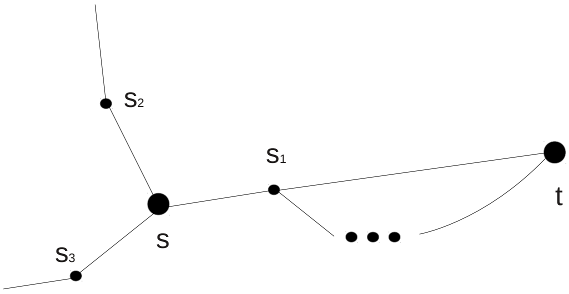


Figure 4.4: No neighboring arcs of s contained in a circle

strictly increasing function, then F is a continuous, strictly increasing function of q. Because of $\lim_{q\to\infty} F(q) = \infty$ and $\lim_{q\to-\infty} F(q) = -\infty$, $F(q) = 0$ has one and only one solution. Note that F has an inverse function F^{-1} which is also a continuous and increasing function. We now set $q := F^{-1}(0)$, the unique solution of CNS(G^o_s) with inflow $Q_s - q$ and $Q_a = q$ is the unique solution of CNS(G_s) with inflow Q_s.

Consider the function F again. Since $f^o_{s_1}$ and p^a both have inverse functions, there exists a function $\bar{f}$ such that

$$Q_s = (f^o_{s_1})^{-1}(-p_a(q)) + q =: \bar{f}(q),$$

where $\bar{f}$ is a continuous, increasing function that maps q to Q_s and has the inverse function $\bar{f}^{-1}$. For q with $F(q) = 0$ it follows

$$Q_s - q = (f^o_{s_1})^{-1}(-p_a(\bar{f}^{-1}(Q_s))) =: \tilde{f}(Q_s).$$

Function $\tilde{f}$ is then a continuous, increasing function that maps Q_s to $Q_s - q$.

From our induction hypothesis, for every node j in G^o_s there exists f^o_j that maps $Q_s - q$ to h_j, then $f_j := f^o_j \circ \tilde{f}$ maps Q_s to h_j which is continuous, decreasing or constant. Analogously, for every arc a in G^o_s there exists g^o_a that maps $Q_s - q$ to Q_a which is continuous, either constant or monotonic. Then the function $g_a := g^o_a \circ \tilde{f}$ has the same property as g^o_a.

For the arc $a = (s, s_1)$ which is not contained in G^o_s, the function $\bar{f}^{-1}$ which maps Q_s to Q_a is continuous, increasing. Again, setting $Q_s = Q^0_s$ yields the unique solution. □

Until now we know that for a given semi-passive subnetwork G_s with $Q_s = Q^0_s \in \mathbb{R}$ and $h_s = H_s \in \mathbb{R}$, where Q_s and H_s are constants, we can solve CNS(G_s) first and then add H_s to h_j for all nodes j to get the unique potential solution. The potential solution is a solution for the subnetwork if it fulfills all constraints (4.2). Otherwise there exists no solution. Note that increasing H_s may turn a violated potential solution into a solution, when Q_s is fixed. Only with appropriate flow at the inflow node G_s we will have at most one solution, this is why we called G_s a semi-passive network.

Assume that all functions f_j and g_a in Theorem 4.4 are known for a semi-passive network G_s. All constraints ((2.1), (2.7), (4.1), (4.2)) related to G_s in the entire MINLP can be replaced by

$$\underbrace{h_s + f_j(Q_s)}_{=h_j} \geqslant H_j^0 \tag{4.3}$$

for all junctions j in G_s, the single constraint for the flow

$$Q_s + Q_t = D. \tag{4.4}$$

and the single constraint for the head

$$h_s - h_t + f_t(Q_s) = 0. \tag{4.5}$$

Note that there are only three variables Q_s, Q_t and h_s in ((4.3), (4.4), (4.5)) which also appear in the constraints related to the remaining graph. To solve the MINLP, we do not have to know the value of Q_a for all arcs a in G_s if there are no other constraints on these variables.

Detection of redundant constraints In the entire MINLP every variable is bounded. Let $[Q_s^{\min}, Q_s^{\max}]$ be the domain of Q_s. For every node j, f_j is a continuous, decreasing or constant function. Hence it follows that

$$f_j(Q_s^{\max}) \leq f_j(Q_s) \leq f_j(Q_s^{\min}).$$

Note that $f_j(Q_s^{\max})$ and $f_j(Q_s^{\min})$ are constants which can be obtained by solving CNS(G_s) with $Q_s = Q_s^{\max}$ or $Q_s = Q_s^{\min}$. The fulfillment of constraint (4.5) implies a lower bound of h_s by

$$\begin{aligned} h_s &= h_t - f_t(Q_s) \\ &\geq h_t - f_t(Q_s^{\min}) \\ &\geq H_t^0 - f_t(Q_s^{\min}). \end{aligned}$$

With $h_s \geq H_s^0$, the constant $\max\{H_s^0, H_t^0 - f_t(Q_s^{\min})\}$ is a lower bound of h_s. With this, for every node $j \in \mathcal{N}\backslash\{s,t\}$, a lower bound of h_j can be found by

$$h_j = h_s + f_j(Q_s) \geq h_s + f_j(Q_s^{\max}) \geq \underbrace{\max\{H_s^0, H_t^0 - f_t(Q_s^{\min})\} + f_j(Q_s^{\max})}_{=:\bar{H}_j}.$$

As $\bar{H}_j$ is a constant, we can compare it with H_j^0. It is clear that for every node $j \in \mathcal{N}_s\backslash\{s,t\}$, the constraint

$$\underbrace{h_s + f_j(Q_s)}_{=h_j} \geqslant H_j^0$$

of type (4.3) is redundant if

$$\bar{H}_j \geq H_j^0.$$

4.3 Symbolic computation

As we discussed at the end of Section 4.2.2, to replace constraints ((2.1), (2.7), (4.1), (4.2)) by constraints ((4.3), (4.4), (4.5)), we need to know the function f_j for every node j in a semi-passive network G_s. Consider CNS(G_s) again. For every arc a there is a function g_a with $Q_a = g_a(Q_s)$, where g_a is continuous, decreasing, increasing or constant. The flow direction through arc a, which corresponds to the sign of Q_a, is said to be decided with inflow $Q_s \in \mathcal{I} := [Q_s^1, Q_s^2]$ if $Q_a \leq 0$ or $Q_a \geq 0$ for all $Q_s \in \mathcal{I}$. Since one of $g_a(Q_s^1)$ and $g_a(Q_s^2)$ is the maximum and the other is the minimum of Q_a, the flow direction is decided if and only if

$$g_a(Q_s^1) \cdot g_a(Q_s^2) \geq 0.$$

With this property we make the following definition.

Definition 4.5 (Flow direction decided domain)
A domain $\mathcal{I} := [Q_s^1, Q_s^2]$ with two constants $Q_s^1, Q_s^2 \in \mathbb{R}$ is said to be a *flow direction decided domain* if for every arc $a \in \mathcal{A}_s$ it holds

$$g_a(Q_s^1) \cdot g_a(Q_s^2) \geq 0.$$

Note that such a flow direction decided domain always exists. Because for any $Q_s^0 \in \mathbb{R}$ a given constant, there exists an $\varepsilon > 0$ so that for every arc a, either $Q_a = g_a(Q_s) \leq 0$ or $Q_a = g_a(Q_s) \geq 0$ for all $Q_s \in [Q_s^0, Q_s + \varepsilon]$, i.e., the sign of Q_a for every arc a is decided for $Q_s \in [Q_s^0, Q_s^0 + \varepsilon]$. Since CNS($G_s$) has a unique solution with fixed Q_s, every NLP solver may find the solution. For a given flow direction decided domain $[Q_s^1, Q_s^2]$, the sign of Q_a, which is $S_a \in \{-1, 1\}$, can be determined as

$$S_a = \begin{cases} 1 & \text{if } g_a(Q_s^1) + g_a(Q_s^2) \geq 0, \\ -1 & \text{otherwise.} \end{cases} \tag{4.6}$$

With the sign assignment $S_a \in \{-1, 1\}$ for every $a \in \mathcal{A}_s$, CNS(G_s) becomes

$$\left.\begin{aligned} \sum_{a=ij\in\mathcal{A}_s} Q_a - \sum_{a=ji\in\mathcal{A}_s} Q_a - D_j = 0 \quad & \text{for } j \in \mathcal{N}_s \setminus \{s,t\} \\ \sum_{a=ij\in\mathcal{A}_s} Q_a - \sum_{a=ji\in\mathcal{A}_s} Q_a + Q_j - D_j = 0 \quad & \text{for } j \in \{s,t\} \\ h_i - h_j - S_a\lambda_a Q_a^2 = 0 \quad & \text{for } a = ij \in \mathcal{A}_s \\ h_s = 0, \quad & \end{aligned}\right\} \tag{SPE$_S(G_s)$}$$

which is a system of polynomial equations SPE$_S(G_s)$. In this system, we regard Q_s, λ_a and S_a as constants or parameter and all Q_a, h_j as variables. Note that Q_t can be eliminated since it depends only on value of Q_s.

Before we solve SPE$_S(G_s)$, we look at the definitions and theorems concerning varieties and ideals. These are taken from the book of Cox et al. [CLO92].

Definition 4.6 (Ideal)
Let k be a field. A subset $I \subset k[x_1, x_2, \dots, x_n]$ of the polynomial ring $k[x_1, x_2, \dots, x_n]$ is an *ideal* if it satisfies:

- $0 \in I$;
- if $f, g \in I$, then $f + g \in I$;
- if $f \in I$ and $h \in \mathbb{R}[x_1, x_2, \dots, x_n]$, then $hf \in I$.

Definition 4.7 (Affine variety and polynomial ideal)
Let k be a field and $f_1, f_2, \dots, f_s$ be polynomials in $k[x_1, x_2, \dots, x_n]$. Then we set

$$\mathbf{V}(f_1, f_2, \dots, f_s) = \{(a_1, \dots, a_n) \in k^n : f_i(a_1, \dots, a_n) = 0 \text{ for all } 1 \le i \le s\}.$$

We call $\mathbf{V}(f_1, f_2, \dots, f_s)$ the *affine variety* defined by $f_1, f_2, \dots, f_s$. Further we set

$$\langle f_1, \dots, f_s \rangle = \left\{ \sum_{t=1}^{s} h_t f_t : h_1, \dots, h_s \in k[x_1, x_2, \dots, x_n] \right\}.$$

Note that $I := \langle f_1, \dots, f_s \rangle$ is an ideal, a proof can be found in the book of Cox et al. [CLO92]. We denote also $\mathbf{V}(I) = \mathbf{V}(f_1, f_2, \dots, f_s)$ as the affine variety defined by $f_1, f_2, \dots, f_s$. From that book we have also the following theorem.

Theorem 4.8 (Elimination and Extension Theory)
Let I be a polynomial ideal with $I = \langle f_1, \dots, f_s \rangle \subset \mathbb{C}[x_1, x_2, \dots, x_n]$ where $\mathbf{V}(I)$ is not an empty set. Assume there exists $p_i(x_i) \in \mathbb{C}[x_i]$ such that $p_i(x_i) \in I$ for one $i \in \{1, \dots, n\}$. It follows then

- *if $\bar{x}_i \in \mathbb{C}$ with $p_i(\bar{x}_i) = 0$, then there exists $(x_1, x_2, \dots, x_n) \in \mathbf{V}(I)$ such that $x_i = \bar{x}_i$;*
- *for any $(x_1, x_2, \dots, x_n) \in \mathbf{V}(I)$, it holds $p_i(x_i) = 0$.*

Furthermore, $\mathbf{V}(I)$ is a finite set if and only if there exists $p_i(x_i) \in \mathbb{C}[x_i]$ with $p_i(x_i) \in I$ for any $i \in \{1, \dots, n\}$.

Note that Theorem 4.8 may not be true if we replace $\mathbb{C}$ by $\mathbb{R}$, since $\mathbb{R}$ is not algebraically closed, e.g., the polynomial $x^2 + 1 \in \mathbb{R}[x]$ has no root in $\mathbb{R}$ but has a root in $\mathbb{C}$. However, the ideal I can be selected so that all $f_i \in \mathbb{R}[x_1, x_2, \dots, x_n] \subset \mathbb{C}[x_1, x_2, \dots, x_n]$.

Consider the system of polynomial equations $\text{SPE}_S(G_s)$ with a given flow direction decided domain $\mathcal{I}$ again. Let $\dot{f}_1, \dot{f}_2, \dots, \dot{f}_s$ be all polynomials in $\text{SPE}_S(G_s)$ which generate an ideal $I = \langle \dot{f}_1, \dots, \dot{f}_s \rangle$. The set $V(I)$ is not an empty set since for parameter $Q_s \in \mathcal{I}$ the corresponding $\text{CNS}(G_s)$ has a solution which is also of solution of $\text{SPE}_S(G_s)$. According to Theorem 4.4, for any fixed given Q_s the polynomial system $\text{SPE}_S(G_s)$ has a unique solution. We handle Q_s as symbolic parameter but not a parameter. Then Theorem 4.8 implies that the corresponding set

$V(I)$ is finite while I is generated by all polynomials in $\text{SPE}_S(G_s)$ Then for every variable h_j or Q_a, polynomials of the form $p_j(h_j)$ or $p_a(Q_a)$ can be found by computing the Gröbner bases with appropriately chosen monomial orderings [Laz83]. There are several software packages, e.g., Maple [Map] and Singular [DGPS12] which can compute Gröbner bases and can check whether the affine variety of a given polynomial system is a finite set. Examples of applications using Maple can be found in [MG94; GDS94; Zac96; RlRmL14; GKZ90].

On the one hand, all of $p_j(h_j)$ and $p_a(Q_a)$ can be regarded as univariate functions which contain symbolic parameter Q_s, every root of $p_j(h_j)$ and $p_a(Q_a)$ is a function that maps Q_s to h_j or to Q_a. On the other hand, for any fixed $Q_s \in \mathcal{I}$, the unique solution for $\text{CNS}(G_s)$ in Theorem 4.4 is a solution for $\text{SPE}_S(G_s)$ as well. It implies that $h_j = f_j(Q_s)$ is a root for $p_j(h_j)$ for all $Q_s \in \mathcal{I}$ and for all nodes j. Similarly, $Q_a = g_a(Q_s)$ is a root for $p_a(Q_a)$ for all $Q_s \in \mathcal{I}$ and for all arcs a. According to Abel-Ruffini theorem [Sko11], there is no algebraic solution to the general univariate polynomial equations of degree five or higher with arbitrary coefficients. It is clear that the roots for a general univariate polynomial can be found if and only if the degree is no more than 4. Assume that the degree of $p_j(h_j)$ is $d_j \leq 4$ (in our instance we have $d_j = 4$), then all d_j roots $r_1, r_2, \ldots, r_{d_j}$ can be found by using Maple or Singular. Note that every r_i can be regarded as a function that maps Q_s to h_j if all other parameters are given as constants. With any fixed $\bar{Q}_s \in \mathcal{I}$, the unique solution of $\text{CNS}(G_s)$ can be found by using any NLP solver, e.g. SCIP.

Let $r_1, \ldots, r_{d_j}$ be the d_j roots of polynomial $p_j(h_j)$. Since any r_j can be regarded as univariate function of Q_s, Theorem 4.4 and Theorem 4.8 imply together that there exists exactly one $\bar{j} \in \{1, \ldots, d_j\}$ such that $f_j(Q_s) = r_{\bar{j}}(Q_s)$ in domain $\mathcal{I}$.

Now we can to find $r_{\bar{j}}$ with an algorithm. Let $\varepsilon \in \mathbb{R}_+$ be the accuracy parameter of the NLP solver and $\bar{h}_j$ be the value of h_j in the solution. If there exists a $\bar{j} \in \{1, 2, \ldots, d_j\}$ such that

$$\left| r_{\bar{j}}(\bar{Q}_s) - \bar{h}_j \right| \leq \varepsilon \text{ and } \left| r_j(\bar{Q}_s) - \bar{h}_j \right| > \varepsilon \text{ for all } j \in \{1, 2, \ldots, d_j\} \backslash \{\bar{j}\},$$

then we can set

$$f_j(Q_s)_{|Q_s \in \mathcal{I}} := r_{\bar{j}}(Q_s)$$

since $r_{\bar{j}}$ is the unique candidate. We call the solution with $Q_s = \bar{Q}_s$ a *distinction solution.* Similarly, all other functions f_j and Q_a in domain $\mathcal{I}$ can be found if the degree of the corresponding polynomial is no more than 4 and a distinction solution can be found.

Let $[Q_s^{\min}, Q_s^{\max}]$ be the domain of Q_s. Recall that all constraints ((2.1), (2.7), (4.1), (4.2)) related to G_s in the entire MINLP can be replaced by constraints ((4.3), (4.4), (4.5)), where only functions f_j appear in these new constraints. We design algorithm 4.1 to compute f_j symbolically for an arbitrarily selected node j. In this algorithm, in the first part, we split the original domain $[Q_s^{\min}, Q_s^{\max}]$ to a set of sub-domains such that in every sub-domain is a flow direction decided domain. After that, we compute f_j in every sub-domain separately. The correctness of the theorem is proved.

Theorem 4.9 (Correctness of Algorithm 4.1)

Algorithm 4.1 finds f_j as defined in Theorem 4.4 if the algorithm terminates with "success".

Algorithm 4.1: Symbolic computation to get f_j

```
Input: A semi-passive subnetwork $G_s$, domain $[Q_s^{\min}, Q_s^{\max}]$ for $Q_s$ and node $j$
Output: Function $f_j$ related to Theorem 4.4 for $Q_s \in [Q_s^{\min}, Q_s^{\max}]$
Initialize $\mathcal{S} := \{[Q_s^{\min}, Q_s^{\max}]\}$ ;
Get the solution of CNS$(G_s)$ with $Q_s = Q_s^{\min}$ and $Q_s = Q_s^{\max}$ using an NLP solver. Let $\acute{Q}_a$
and $\grave{Q}_a$ be the value of $Q_a$ for every arc $a$ in the solution with $Q_s = Q_s^{\min}$ and $Q_s = Q_s^{\max}$,
respectively.
for $a \in \mathcal{A}_s$ do
    if $\acute{Q}_a \cdot \grave{Q}_a < 0$ then
        Solve CNS$(G_s)$ with additional constraint $Q_a = 0$, let $\bar{Q}_s$ be the value of $Q_s$ in the
        unique solution;
        Get the interval $\mathcal{I}_a \in \mathcal{S}$ which is $[Q_1, Q_2)$ (or $[Q_1, Q_2]$) and contains $\bar{Q}_s$;
        if $\bar{Q}_s \in (Q_1, Q_2)$ then
            Replace $\mathcal{I}_a$ by $[Q_1, \bar{Q}_s)$ and $[\bar{Q}_s, Q_2)$ (or $[\bar{Q}_s, Q_2]$) in $\mathcal{S}$ ;
        end
    end
end
for every $\mathcal{I}_i \in \mathcal{S}$ do
    Get sign assignment $S$ according to (4.6) for $Q_s \in \mathcal{I}_i$, construct the corresponding
    SPE$_S(G_s)$ ;
    Get $p_j(h_j)$ by computing Gröbner bases ;
    if $p_j(h_j)$ does not exist then
        Report "not a finite set" and stop
    end
    Let $d_j$ be the degree ;
    if $d_j > 4$ then
        Report "degree too high" and stop
    end
    Compute roots $r_1, \ldots, r_{d_j}$;
    while true do
        Choose a $\bar{Q}_s \in \mathcal{I}_i$ (randomly) and compute the solution of CNS$(G_s)$ with $Q_s = \bar{Q}_s$;
        if the solution is a distinction solution then
            Get $\bar{j}$ and set $f_j(Q_s)_{|Q_s \in \mathcal{I}_i} := r_{\bar{j}}(Q_s)$;
            break
        end
    end
end
Report "success" and return $f_j$
```

Proof. The algorithm initializes a set of domains which contains only the initial domain $[Q_s^{\min}, Q_s^{\max}]$. The domain $[Q_s^{\min}, Q_s^{\max}]$ may not be a flow direction decided domain. It is not, the sign of some Q_a has been changed. Due to the monotonicity of g_a, the sign can be changed at most once, which is equivalent to $\acute{Q}_a \cdot \grave{Q}_a < 0$, where $\acute{Q}_a$ and $\grave{Q}_a$ are defined and found in Algorithm 4.1.

The first for-loop splits $[Q_s^{\min}, Q_s^{\max}]$ into several domains which are disjoint, flow direction decided and their union is exactly $[Q_s^{\min}, Q_s^{\max}]$.

Consider any arc a with $\acute{Q}_a \cdot \grave{Q}_a < 0$. Note that g_a has an inverse function which maps Q_a to Q_s, CNS(G_s) has a unique solution if we fix $Q_a = 0$. The solution can be computed by any NLP solver. Let $\bar{Q}_s$ be the value of Q_s in the unique solution such that $Q_a = g_a(\bar{Q}_s) = 0$. There will be only one interval $\mathcal{I}_a \in \mathcal{S}$ which contains $\bar{Q}_s$. After splitting $\mathcal{I}_a$ into two intervals according to $\bar{Q}_s$, the sign of Q_a is decided for all domains in $\mathcal{S}$ for Q_s.

The first for-loop terminates and $\mathcal{S}$ contains at most $|\mathcal{A}_s|$ domains.

The second for-loop computes the function f_j. In every iteration, the function f_j for $Q_s \in \mathcal{I}_i$ is computed separately. The termination with "success" indicates that the univariate polynomial $p_j(h_j)$ exists, its degree d_j is not more than 4 and a distinction solution can be found for every iteration. By composition of all $f_j(Q_s)_{|Q_s \in \mathcal{I}_i}$ we get the continuous function f_j for $Q_s \in [Q_s^{\min}, Q_s^{\max}]$. □

4.4 Computational results

This section presents results of our computational experiments. For our experiments we ran SCIP 5.0.1 [Sci] with SoPlex 3.1.0 [Sop] as LP solver, CppAD 20160000.1 [Cpp] as expression interpreter for evaluating nonlinear functions for our nonconvex NLPs and MINLPs, Ipopt 3.12 [Ipo] as NLP solver. We used Maple 16 for all symbolic computations. SCIP was run with default settings and a time limit of one hour and a gap limit of 10^{-5}. We conducted the experiments on an Intel(R) Xeon(R) CPU E5-1620 v4 with 3.5 GHz, 10240 KB cache, and 128 GB RAM.

Our computational experiments are based on the instance n25p22a18 which has been introduced in Section 3.3.1 while Figure 3.2 shows the network topology.

Figure 4.5 shows a real-world semi-passive subnetwork which is contained in network n25p22a18. For this subnetwork, without loss of generality, we define *junction01* to be the node s and then *junction13* is the node t. In this subnetwork, there are only two junctions with a nonzero demand D_j. Let the demands be D_1 and D_2. For this subnetwork, as we discussed before, we can replace constraints ((2.1), (2.7), (4.1), (4.2)) which are contained in the original MINLP by constraints ((4.3), (4.4), (4.5)). First we can detect if some of the constraints of type (4.3) are redundant, using the strategy described at the end of Section 4.2.2. In this subnetwork with all given demand combinations (D_1, D_2), for every node $j \in \mathcal{N}_s \setminus \{s, t\}$, the constraint (4.3) is a redundant constraint. For the replacement we need only to get the function f_t. For this subnetwork with all demand combinations, f_t can be found successfully by using Algorithm 4.1 while Maple 16 has been used for all symbolic computations and SCIP has been used as a global solver. A part of f_t for a demand combination is shown in Figure 4.6.

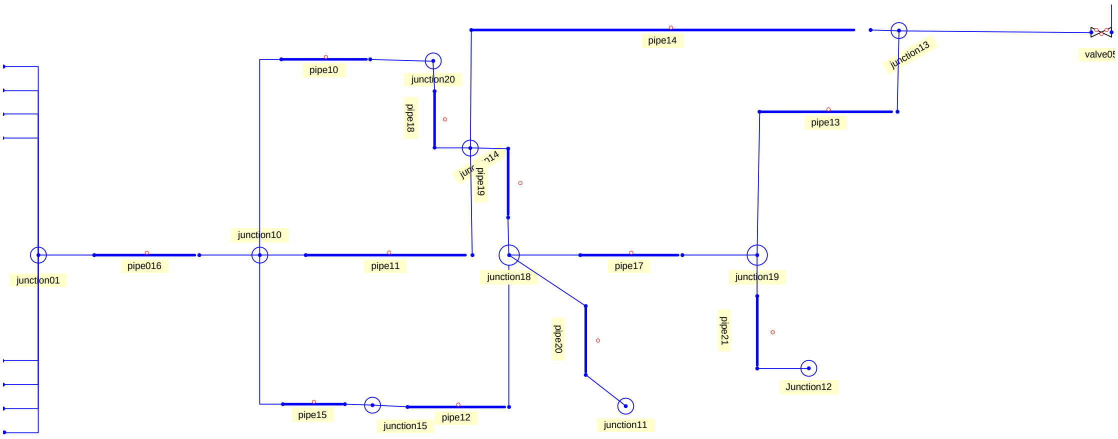
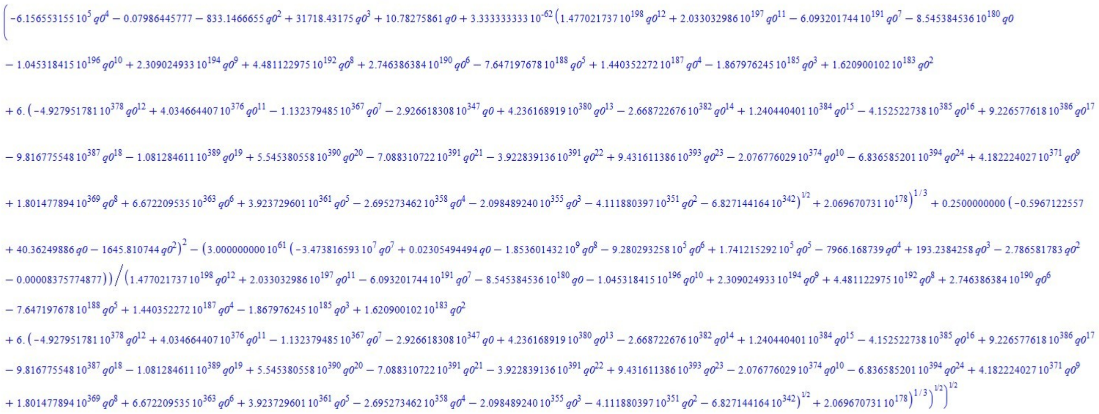


Figure 4.5: A semi-passive subnetwork contained in water supply network `n25p22a18`



$$\Big(-6.156553155\,10^{5}\,q0^{4} - 0.07986445777 - 833.1466655\,q0^{2} + 31718.43175\,q0^{3} + 10.78275861\,q0 + 3.333333333\,10^{-62}\big(1.477021737\,10^{198}\,q0^{12} + 2.033032986\,10^{197}\,q0^{11} - 6.093201744\,10^{191}\,q0^{7} - 8.545384536\,10^{180}\,q0$$

$$- 1.045318415\,10^{196}\,q0^{10} + 2.309024933\,10^{194}\,q0^{9} + 4.481122975\,10^{192}\,q0^{8} + 2.746386384\,10^{190}\,q0^{6} - 7.647197678\,10^{188}\,q0^{5} + 1.440352272\,10^{187}\,q0^{4} - 1.867976245\,10^{185}\,q0^{3} + 1.620900102\,10^{183}\,q0^{2}$$

$$+ 6.\big(-4.927951781\,10^{378}\,q0^{12} + 4.034664407\,10^{376}\,q0^{11} - 1.132379485\,10^{367}\,q0^{7} - 2.926618308\,10^{347}\,q0 + 4.236168919\,10^{380}\,q0^{13} - 2.668722676\,10^{382}\,q0^{14} + 1.240440401\,10^{384}\,q0^{15} - 4.152522738\,10^{385}\,q0^{16} + 9.226577618\,10^{386}\,q0^{17}$$

$$- 9.816775548\,10^{387}\,q0^{18} - 1.081284611\,10^{389}\,q0^{19} + 5.545380558\,10^{390}\,q0^{20} - 7.088310722\,10^{391}\,q0^{21} - 3.922839136\,10^{391}\,q0^{22} + 9.431611386\,10^{393}\,q0^{23} - 2.076776029\,10^{374}\,q0^{10} - 6.836585201\,10^{394}\,q0^{24} + 4.182224027\,10^{371}\,q0^{9}$$

$$+ 1.801477894\,10^{369}\,q0^{8} + 6.672209535\,10^{363}\,q0^{6} + 3.923729601\,10^{361}\,q0^{5} - 2.695273462\,10^{358}\,q0^{4} - 2.098489240\,10^{355}\,q0^{3} - 4.111880397\,10^{351}\,q0^{2} - 6.827144164\,10^{342}\big)^{1/2} + 2.069670731\,10^{178}\big)^{1/3} + 0.2500000000\,\big(-0.5967122557$$

$$+ 40.36249886\,q0 - 1645.810744\,q0^{2}\big)^{2} - \big(3.000000000\,10^{61}\,\big(-3.473816593\,10^{7}\,q0^{7} + 0.02305494494\,q0 - 1.853601432\,10^{9}\,q0^{8} - 9.280293258\,10^{5}\,q0^{6} + 1.741215292\,10^{5}\,q0^{5} - 7966.168739\,q0^{4} + 193.2384258\,q0^{3} - 2.786581783\,q0^{2}$$

$$- 0.00008375774877\big)\big)\Big/\big(1.477021737\,10^{198}\,q0^{12} + 2.033032986\,10^{197}\,q0^{11} - 6.093201744\,10^{191}\,q0^{7} - 8.545384536\,10^{180}\,q0 - 1.045318415\,10^{196}\,q0^{10} + 2.309024933\,10^{194}\,q0^{9} + 4.481122975\,10^{192}\,q0^{8} + 2.746386384\,10^{190}\,q0^{6}$$

$$- 7.647197678\,10^{188}\,q0^{5} + 1.440352272\,10^{187}\,q0^{4} - 1.867976245\,10^{185}\,q0^{3} + 1.620900102\,10^{183}\,q0^{2}$$

$$+ 6.\big(-4.927951781\,10^{378}\,q0^{12} + 4.034664407\,10^{376}\,q0^{11} - 1.132379485\,10^{367}\,q0^{7} - 2.926618308\,10^{347}\,q0 + 4.236168919\,10^{380}\,q0^{13} - 2.668722676\,10^{382}\,q0^{14} + 1.240440401\,10^{384}\,q0^{15} - 4.152522738\,10^{385}\,q0^{16} + 9.226577618\,10^{386}\,q0^{17}$$

$$- 9.816775548\,10^{387}\,q0^{18} - 1.081284611\,10^{389}\,q0^{19} + 5.545380558\,10^{390}\,q0^{20} - 7.088310722\,10^{391}\,q0^{21} - 3.922839136\,10^{391}\,q0^{22} + 9.431611386\,10^{393}\,q0^{23} - 2.076776029\,10^{374}\,q0^{10} - 6.836585201\,10^{394}\,q0^{24} + 4.182224027\,10^{371}\,q0^{9}$$

$$+ 1.801477894\,10^{369}\,q0^{8} + 6.672209535\,10^{363}\,q0^{6} + 3.923729601\,10^{361}\,q0^{5} - 2.695273462\,10^{358}\,q0^{4} - 2.098489240\,10^{355}\,q0^{3} - 4.111880397\,10^{351}\,q0^{2} - 6.827144164\,10^{342}\big)^{1/2} + 2.069670731\,10^{178}\big)^{1/3}\big)^{1/2}\Big)^{1/2}$$

Figure 4.6: A part of function f_t

Unfortunately, these f_t can not be used by our MINLP solver SCIP efficiently. However, we can use polynomials of Q_s to fit f_t by using, e.g., the method of least squares. Since SCIP has nice performance to handle quadratic constraints [BHV12], we first try quadratic polynomials in the form of $\tilde{f}_t = c_0 + c_1 Q_s + c_2 Q_s^2$ to fit f_t. The computed result is shown in Table 4.1. Maple 16 needs less than 10 seconds to get all parameters (c_0, c_1, c_2).

1

Table 4.1: Polynomial fitting of the function f_t and the errors.

D_1	D_2	c_0	c_1	c_2	max error (-)	max error (+)
90	25.2	0.067	-4.466	123.711	-0.0006	0.0005
90	36	0.076	-4.745	123.711	-0.0005	0.0005
90	43.2	0.083	-4.931	123.711	-0.0005	0.0005
90	46.8	0.087	-5.024	123.712	-0.0005	0.0005
90	86.4	0.134	-6.047	123.713	-0.0009	0.0010
90	111.6	0.170	-6.698	123.715	-0.0014	0.0017
90	115.2	0.176	-6.791	123.715	-0.0015	0.0018
90	122.4	0.188	-6.978	123.716	-0.0016	0.0020
90	126	0.193	-7.071	123.716	-0.0016	0.0020
90	133.2	0.206	-7.257	123.717	-0.0018	0.0022
90	147.6	0.232	-7.630	123.719	-0.0021	0.0029

After that, the maximal error of the polynomial fitting can be computed by solving the NLPs

$$\begin{aligned} \min/\max \quad & c_0 + c_1 Q_s + c_2 Q_s^2 - h_t \\ \text{s.t.} \quad & \text{CNS}(G_s) \\ & Q_s \in [Q_s^{\min}, Q_s^{\max}] \end{aligned}$$

The maximal errors are shown in Table 4.1. The total maximal error is less than 0.003 meter which can be ignored with comparison to the measurement errors. As a result, these polynomials can replace f_t in our further computations.

The test instance is given with a demand of 24 hours. Let $P[i, j]$ denote the modeled operation problem from hour i to j with $0 \leq i < j \leq 24$. We call $[i, j]$ the *planning period*. In the following, we want to compare the original MINLP model (2.26) which we have introduced in Section 2.1 and the simplified MINLP obtained by replacing constraints ((2.1), (2.7), (4.1), (4.2)) by constraints ((4.3), (4.4), (4.5)) in the original MINLP. Since the new MINLP has less variables and constraints, we call it *reduced* MINLP. Table 4.2 shows how the replacement helps to reduce the size of presolved MINLPs by SCIP in number of variables "vars", binary variables "bin", number of constraints "cons" and number of nonlinear constraints "nlin" for selected planning periods.

Finally, Table 4.3 presents the computational results for the 24 original MINLPs and the corresponding simplified and approximated MINLPs which model the operation problems for the first i hours with $i = 1, \ldots, 24$. For every MINLP we set a time limit of an hour. In column "gap (time)" the time has been displayed if the gap limit has been reached within an hour; otherwise the time limit has been reached, only the current gap needs to be displayed. From the table we see that there are more simplified MINLPs which can be solved to optimality within

Table 4.2: Problem sizes for the original and reduced MINLPs for water supply network `n25p22a18`

MINLP	without replacement				with replacement			
	vars	bin	cons	nlin	vars	bin	cons	nlin
P[0, 1]	101	14	135	39	80	13	113	32
P[0, 2]	222	29	304	79	180	27	260	65
P[0, 12]	1442	179	1994	479	1190	168	1730	395
P[0, 24]	2905	359	4021	958	2389	336	3481	790
MINLP	without replacement				with replacement			
	vars	bin	cons	nlin	vars	bin	cons	nlin

one hour than original MINLPs. If both cannot be solved to optimality, the gap for the simplified MINLP is much less than the gap for the original MINLP.

Table 4.3: Detailed computational results with the original and reduced MINLPs for water supply network n25p22a18, computed by SCIP 5.0.1

MINLP	original			with replacement		
	primal	dual	(time) gap	primal	dual	(time) gap
P[0, 1]	50.77	50.77	(0.44) 0	50.77	50.77	(0.1) 0
P[0, 2]	125.83	125.83	(1.42) 0	125.83	125.83	(0.96) 0
P[0, 3]	185.2	185.2	(5.61) 1.68e-08	185.2	185.2	(2.52) 1.23e-08
P[0, 4]	233.55	233.55	(350.97) 9.02e-08	233.55	233.55	(64.19) 2.35e-08
P[0, 5]	282.44	282.44	(20.23) 0	282.44	282.44	(547.32) 9.77e-08
P[0, 6]	341.15	341.15	(51.53) 0	341.15	341.15	(14.63) 7.24e-08
P[0, 7]	392.85	392.85	(26.57) 0	392.85	392.85	4.658e-07
P[0, 8]	448.79	448.79	(376.7) 3.91e-08	448.79	448.79	2.575e-07
P[0, 9]	534.48	534.48	(176.51) 0	534.48	534.48	(1056.9) 0
P[0, 10]	671.47	671.47	(465.89) 0	671.47	671.47	(230.46) 0
P[0, 11]	768.8	768.8	(1396.85) 0	768.8	768.8	(78.44) 8.05e-08
P[0, 12]	870.98	870.96	2.37186e-05	870.98	870.98	(112) 0
P[0, 13]	935.85	898.91	0.0411002	935.85	935.85	(578.07) 0
P[0, 14]	991.83	960.96	0.0321309	991.83	991.83	(183.54) 0
P[0, 15]	1055.73	949.15	0.112285	1055.73	1055.73	(553.88) 0
P[0, 16]	1130.53	1013.99	0.114935	1130.53	1130.53	(2535.95) 9.76e-08
P[0, 17]	1228.14	1128.73	0.0880757	1228.14	1228.14	(438.49) 0
P[0, 18]	1352.26	1158.17	0.167585	1352.09	1351.34	0.000553345
P[0, 19]	1401.82	1222.86	0.146347	1401.82	1400.89	0.000662269
P[0, 20]	1453.89	1280.42	0.135482	1453.89	1447.68	0.00428762
P[0, 21]	1547.68	1339.51	0.155409	1533.84	1514.23	0.0129535
P[0, 22]	1982.49	1383.74	0.4327	1701.89	1663.46	0.0231068
P[0, 23]	1909.44	1547.66	0.2338	1883.77	1818.38	0.0359563
P[0, 24]	2593.78	2227.87	0.16424	2590.18	2518	0.0286644

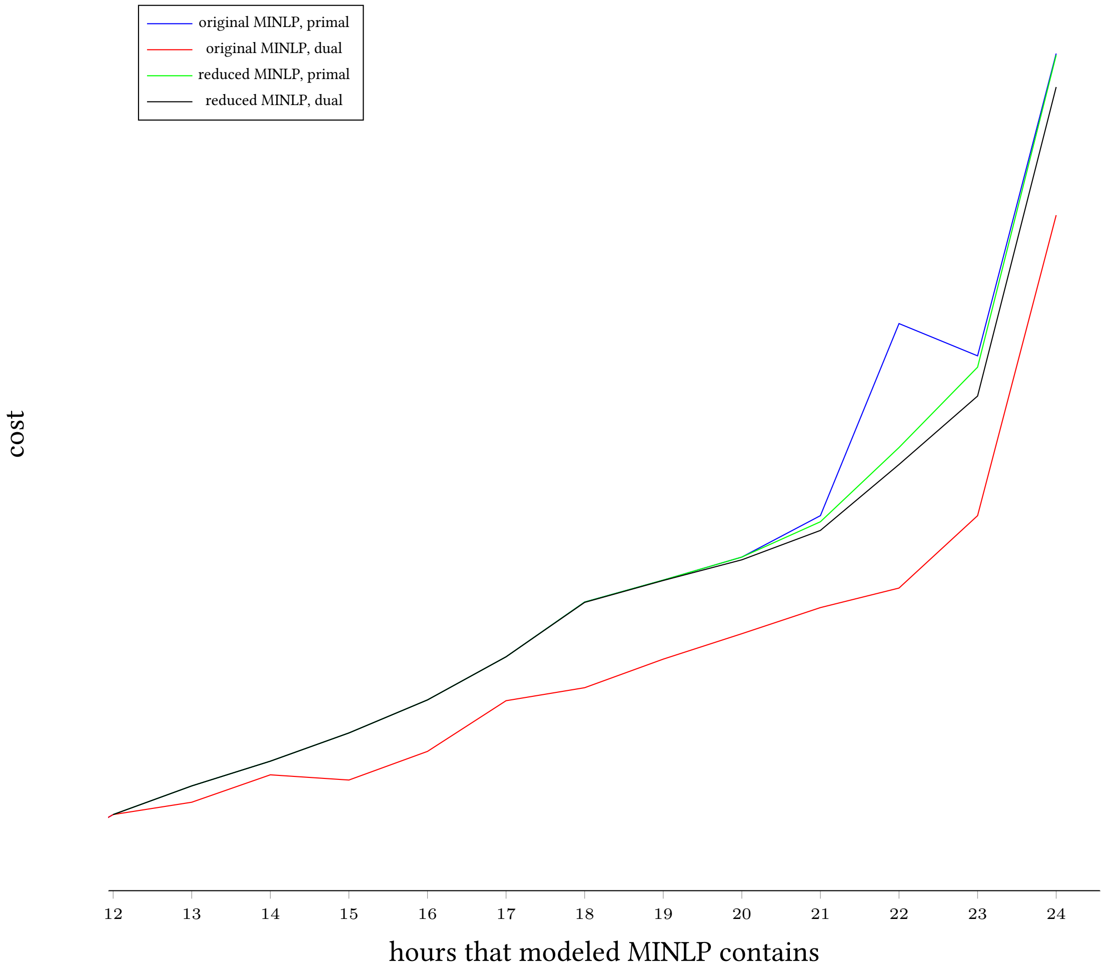


Figure 4.7: Comparison of primal and dual bounds by solving the original and reduced MINLPs for water supply network `n25p22a18`, computed by SCIP 5.0.1

Chapter 5

Convex Hull of Graphs of Polynomial Functions

In Chapter 3 we explained that the MINLP solver SCIP can handle nonlinear terms like x^2 or $x_1 \cdot x_2$. Based on that, SCIP may handle all types of polynomial functions since we may substitute all the nonlinear terms into the form x^2 or $x_1 \cdot x_2$. However, the resulting outer approximation could be very loose. We will show this by computations at the end of this chapter. Vigerske [Vig12] gives a list of polynomial functions which can be directly handled by SCIP. For a further instance which we will present later in this chapter, we use bivariate polynomial functions to approximate energy cost of pumps. These polynomial functions do not fulfill the conditions mentioned in [Vig12]. Thus, SCIP does not generate outer approximation for them directly. These functions appear however directly in the objective and bad outer approximations lead to bad dual bounds.

In our MINLP algorithm we only add linear constraints into the LP relaxation such that the original feasible region is contained in the corresponding outer-approximation. For constraints with polynomial functions the feasible region corresponds to the graph of them. The research motivation is that linear constraints getting directly by observing the feasible region of bivariate polynomial constraints will be tighter than those getting for substituted nonlinear constraints.

Note that variables appearing in nonlinear terms in MINLP applications are usually box-constrained. In this chapter, we investigate first the convex hull of graphs of polynomial functions over a polytope. This convex hull is usually formed by infinitely many halfspaces. For these halfspaces we define and show which of them are “efficient” and which are not. This study is for general polynomial functions of dimension n.

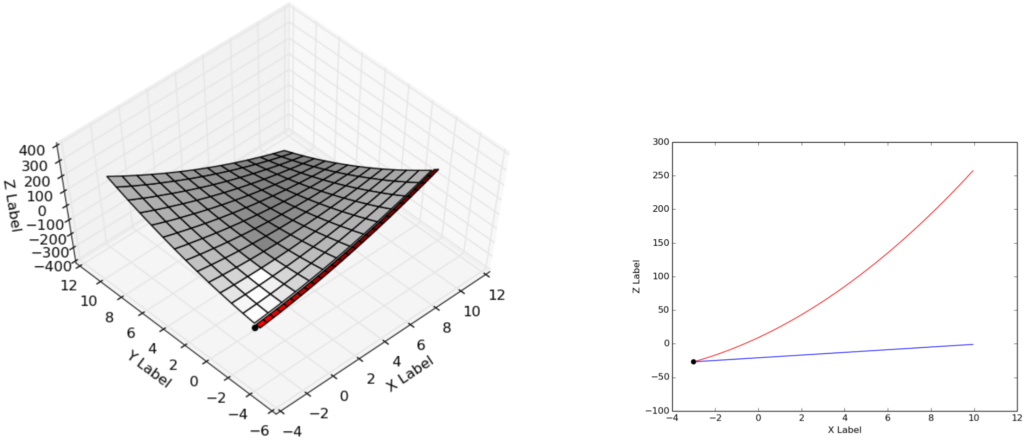
Based on the complete theory we go back to our application and design algorithms to compute “efficient” halfspaces and add them into our MINLP. Computations show that these halfspaces improve the dual bounds significantly.

5.1 Literature survey

Tawarmalani and Sahinidis [TS04; TS05] have shown that for the usual global branch-and-bound approach for nonconvex MINLP, a construction of sharp relaxations is essential. Recalling the definition, the tightest convex outer-approximation of a nonconvex set is obtained by the convex hull relaxation. A detailed theoretical observation on such relaxations has been achieved in the 1970s [Roc70], and more recently in the current century [HUL01; Ber09]. Typically, these convex relaxations are obtained by replacing each nonconvex term individually by convex underestimators or concave overestimators. Plenty of computational results showed that the tighter the convex relaxation is, the tighter the bound is. Consider a nonconvex feasible region defined by a nonconvex function over a convex compact domain with a function type that is frequently used for formulating MINLPs. By definition, finding the convex hull of the nonconvex graph is equivalent to finding the convex and concave envelope of the function. There is very limited availability of formulae for these envelopes due to the fact that the standard representation of envelopes is equivalent to a nonconvex optimization problem that is intractable, in general [JMW08]. Despite of the strong requirement of the quality of outer-approximations, very little literature can be found, even only for general polynomial functions.

However, instead of handling general functions directly, there are several works on the construction of the convex hulls for problems with special structures. Jach et al. [JMW08] have shown how to reduce the problems of determining a convex envelope to lower-dimensional optimization problems when the underlying function is indefinite and $(n-1)$-convex. This allows to describe the convex envelope of a variety of two-dimensional functions. Boland et al. [BDKMR17] have investigated the gap between the McCormick relaxation and convex hull for bilinear functions. For bivariate functions with a fixed convexity behavior, Ballerstein et al. [BMV13] have shown how to construct underestimators. Lundell et al. [LW09] have given strategies to find convex underestimation for signomial functions. For supermodular functions and disjunctive functions, Tawarmalani et al. [TRX13] have derived explicit characterizations of their corresponding envelopes. In addition, convex envelopes of

- multilinear functions over a unit hypercube and over special discrete sets [She97],
- lower semi-continuous functions defined over compact convex sets [KS13],
- monomials of odd degree [LP03],
- edge-concave functions [MF05],
- low-dimensional quadratic forms [AB10],
- bilinear, fractional and other bivariate functions over general polytopes [Loc16],
- some quadratic functions over the n-dimensional unit simplex [Loc15],
- bivariate functions over polytopes [LS14],
- lower semi-continuous functions [TS02b],
- products of convex and component-wise concave functions [KS12],
- trilinear monomials with mixed sign domains [MF04]



(a) The graph of a polynomial function and a piece of the boundary

(b) A subgraph on the boundary

Figure 5.1: Investigating the boundary of the graph of a polynomial function

are presented, respectively. In this chapter, we focus on the characteristics of the convex hull of graphs of general polynomial functions over a polytope.

5.2 Basic ideas of this chapter

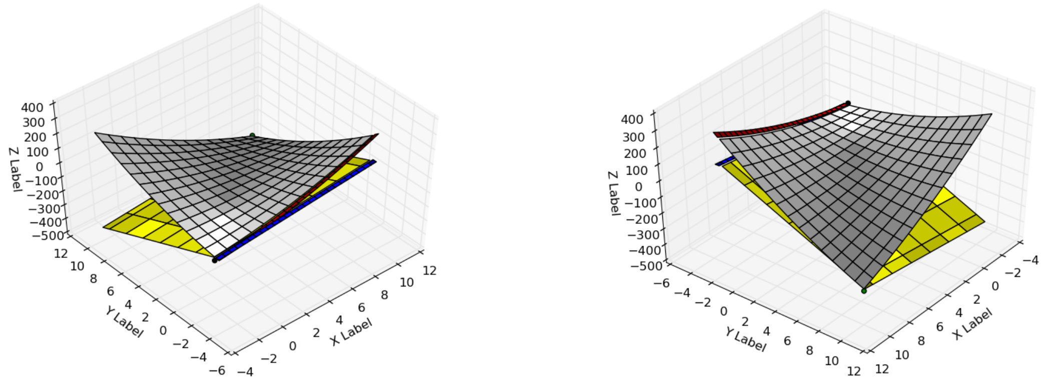
Consider the following example in $\mathbb{R}^3$. Let $f(x, y) = x^2 - 5xy + y^2$ be a polynomial function over the domain $X = \{(x, y) \in [-3, 10] \times [-3, 10]\}$. For the constraint $z = x^2 - 5xy + y^2$, the feasible region, denoted by $\mathcal{S}$, is shown in Figure 5.1a which corresponds to the graph of f over domain X. Recalling our MINLP algorithms, we add linear constraints to strengthen the LP relaxation. For any hyperplane H in $\mathbb{R}^3$ defined in the form $\{(x, y, z) \mid z = ax + by + c\}$ with constants $a, b, c \in \mathbb{R}$, H is said to be a linear underestimator to f over X if

$$\mathcal{S} = \{(x, y, z) \mid z = f(x, y), (x, y) \in X\} \subset \underbrace{\{(x, y, z) \mid z \geq ax + by + c\}}_{\text{downward closed halfspace to } H}.$$

Graphically, it means that the corresponding downward closed halfspace completely contains the graph of f over X.

In contrast to general MINLP algorithms, we want to find such linear underestimators directly. They are expected to strengthen the LP relaxation. The intuition is that we only want to consider such hyperplanes H that support the graph, otherwise we can move it upwardly until the new generated hyperplane intersects the graph.

In other words, we say a linear underestimators H is *below* (see Definition 5.16) the graph $\mathcal{S}$. Thus H is said to be *valid*.



(a) The view of the graph and a linear underestimator

(b) The view from the other side

Figure 5.2: A linear underestimator which supports two boundary points of the graph of a polynomial function

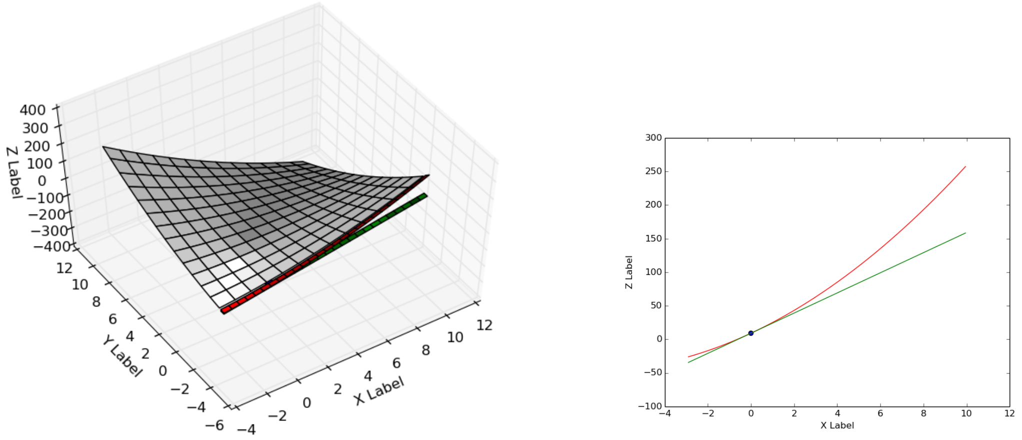
To find linear underestimators H, we study the intersection points $H \cap \mathcal{S}$. After a series of preliminary definitions in Section 5.3.1, we define locally and globally convex points in Section 5.3.2. A point (x^0, y^0, z^0) on the graph is said to be locally convex if there exists $H \ni (x^0, y^0, z^0)$ and H is below the graph of f over a neighborhood of (x^0, y^0, z^0). A point (x^1, y^1, z^1) on the graph is said to be globally convex if there exists $H \ni (x^1, y^1, z^1)$ and H is below $\mathcal{S}$.

The hyperplane $H^t = \{(x, y, z) \mid z = 9x - 30y - 90\}$, shown as the yellow hyperplane in Figure 5.2, can be verified to be a linear underestimator for f over X. The hyperplane H^t intersects $\mathcal{S}$ in two points $(-3, -3, -27)$ and $(10, 10, -300)$. Hence $(-3, -3, -27)$ and $(10, 10, -300)$ both are globally convex points. Note that they both are boundary points of $\mathcal{S}$.

Consider further a point (x^0, y^0, z^0) such that the corresponding domain point (x^0, y^0) is an interior point of X. As we will show in Section 5.3.2, to check if (x^0, y^0, z^0) is globally convex, we need only to check if the corresponding tangent plane is below $\mathcal{S}$. However, in practice, it is quite hard to find those globally convex points such that the corresponding domain points are interior points of X. In addition, the property of global convexity usually depends on the domain. On the one hand, any locally convex point may become globally convex if the domain size is small enough. On the other hand, a globally convex point with respect to the current domain could be only locally convex for a larger domain. Note that in the example above f is neither convex nor concave over X.

Now we move our attention to those globally convex points for which the corresponding domain points are on the boundary of X. Consider the subgraph with restriction $y = -3$, which is presented as

$$\{(x, y, z) \mid z = f(x, y), -3 \leq x \leq 10, y = -3\}.$$



(a) Subtangent plane

(b) Globally convex boundary point in $\mathbb{R}^2$

Figure 5.3: Example for a globally convex boundary point

This subgraph is shown as the red curve in Figure 5.1a. Since $y = -3$ is satisfied for any point in the red subgraph, after projecting the space $\{(x, -3, z)\} \subset \mathbb{R}^3$ to the space $\{(x, z)\} \subset \mathbb{R}^2$, we get an isomorphic two-dimensional curve in $\mathbb{R}^2$

$$\{(x, z) \mid z = f(x, -3) = x^2 + 15x + 9 =: \tilde{f}(x), -3 \leq x \leq 10\}.$$

In general, we show at the beginning of Section 5.3.3 that certain subgraphs on the boundary can be projected to an isomorphic graph in a space with lower dimension. The one-dimensional curve is shown as the red curve in Figure 5.1b. Note that the corresponding function $\tilde{f}$ is a univariate function. Fortunately, the study of the convexity of univariate functions is much easier than that for bivariate functions. In the example $\tilde{f}$ has domain $[-3, 10]$ and is a convex function over $[-3, 10]$.

According to the definition of globally convex points, any point $x^* \in [-3, 10]$ in Figure 5.1b is globally convex in the projected space $\mathbb{R}^2$. Theorem 5.12 implies that for any such x^*, because x^* is globally convex in the projected space, the boundary point $(x^*, -3)$ in X is also globally convex in the original space. This means there exists a hyperplane $H \ni (x^*, -3, f(x^*, -3))$ and H is below $\mathcal{S}$. Consider the case $x^* = 0$. Then $(0, -3, 9)$ is a globally convex point. Figure 5.3b shows that in the projected space $\mathbb{R}^2$, the tangent plane, shown as the green line, is the unique underestimator. The corresponding line in $\mathbb{R}^3$, shown as the green line in Figure 5.3a, is then $\{(x, -3, 15x + 9) \mid x \in \mathbb{R}\}$ which is defined as subtangent plane in Section 5.3.3. Corollary 5.13 implies that every linear underestimator H with $H \ni (0, -3, 9)$ satisfies

$$H \supset \{(x, -3, 15x + 9) \mid x \in \mathbb{R}\},$$

which means any valid hyperplane which contains $(0, -3, 9)$ always contains the green line.

The blue line $\{(x,-3,9x) \mid x \in \mathbb{R}\}$ in Figure 5.2a is a subtangent plane on $(-3,-3,-27)$, as defined in Section 5.3.3. We can verify that the yellow hyperplane is the affine hull of the blue line and the point $(10,10,-300)$, i.e.,

$$H^t = \{(x,y,z) \mid z = 9x - 30y - 90\} = \text{aff}\,\{\{(x,-3,9x) \mid x \in \mathbb{R}\}, \{(10,10,-300)\}\}\,.$$

For any point $(10,10,z^1)$ with $z^1 < -300$, we can also verify that

$$H^l = \text{aff}\left\{\{(x,-3,9x) \mid x \in \mathbb{R}\}, \{(10,10,z^1)\}\right\}$$

is also a linear underestimator. By comparing H^t and H^l we have

$$H^t \cap \mathcal{S} = \{(-3,-3,-27),(10,10,-300)\} \supsetneq \{(-3,-3,-27)\} = H^l \cap \mathcal{S}.$$

From the intuition, we prefer H^t since the resulting relaxation is tighter. For this purpose we define tight and loose hyperplanes in Section 5.3.4. In general, a valid hyperplane H^l is definitely loose if there exists another valid hyperplane H^t which preserves all intersection points and intersects in additional point(s) with $\mathcal{S}$, which means

$$(H^t \cap \mathcal{S}) \supsetneq (H^l \cap \mathcal{S}).$$

This is a sufficient but not necessary condition for loose hyperplanes. Using Lemma 5.26 in Section 5.3.4 we verify that the yellow hyperplane in Figure 5.2a is a tight hyperplane.

After that, in Section 5.3.5 we prove for every loose hyperplane H^l that there exists a tight hyperplane H^t that preserves intersection points with

$$(H^t \cap \mathcal{S}) \supset (H^l \cap \mathcal{S}).$$

We call the corresponding halfspaces tight or loose halfspaces. Note that in the example above we have $H^t \cap H^l = \{(x,-3,9x) \mid x \in \mathbb{R}\}$ which is the blue line in Figure 5.2a. Graphically, we can rotate H^l around the blue line as axis to generate H^t. The rotation approach is the basic idea of a few proofs in this section.

Finally, in Section 5.3.6, we prove that to form the convex hull of $\mathcal{S}$ using halfspaces, we only need tight hyperplanes. In other words, any loose hyperplane is proved to be redundant.

In Section 5.3 we only include theoretical results. We cannot use them to solve MINLP directly. In Section 5.4 we develop algorithms to compute tight hyperplanes for the graph of bivariate polynomial functions with degree up to 3 over a polygon in $\mathbb{R}^2$. Note that the domain does not have to be box-constrained. In the algorithms, we first find all globally convex domain points on the boundary. This is very tractable since we only need to find globally convex points in the graph of univariate polynomial functions with degree 3 over a closed interval in $\mathbb{R}$. Based on those globally convex domain points, the algorithms find a series of tight halfspaces. Computations in Section 5.5 show that these tight halfspaces improve the dual bounds significantly.

5.3 Convex hull of graphs of polynomial functions over a polytope

5.3.1 Preliminary definitions

Let $m, n \in \mathbb{N}$ and $X \subset \mathbb{R}^n$ be a polytope defined by the intersection of finitely many halfspaces, i.e., $X := \{\mathbf{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n \mid a_j^T \mathbf{x} \leq b_j, j = 1, \ldots, m\}$, where $a_j \in \mathbb{R}^n, b_j \in \mathbb{R}$ for $j = 1, \ldots, m$. Then, for a given polynomial function $f : X \to \mathbb{R}$ with $f \in \mathbb{R}[\mathbf{x}]$, the image of the function $F : X \to \mathbb{R}^{n+1}, \mathbf{x} \mapsto (\mathbf{x}, f(\mathbf{x}))$ defines the graph of f. Note that some of definitions and properties are also suitable for general differentiable functions. But in this chapter we focus on polynomial functions.

In this section, we study the convex hull of the set

$$\mathcal{S} := \{(\mathbf{x}, z) \mid z = f(\mathbf{x}),\ \mathbf{x} \in X\} \subset \mathbb{R}^{n+1}.$$

Define projection functions $\pi_{\mathbf{x}} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n, (\mathbf{x}, z) \mapsto \mathbf{x}$ and $\pi_z : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}, (\mathbf{x}, z) \mapsto z$. For every point $(\mathbf{x}, f(\mathbf{x})) \in \mathcal{S}, \mathbf{x} = \pi_{\mathbf{x}}((\mathbf{x}, f(\mathbf{x})))$ is the corresponding domain point.

Theorem 5.1
Let $S \subset \mathbb{R}^{n+1}$ be a compact set. Then the convex hull of S is a compact set which is the intersection of all closed halfspaces containing S.

Proof. This can be directly derived from Corollary 5.33 and 5.83 of the book [AB06]. □

Without loss of generality, we assume X be to full-dimensional, otherwise we can reduce the dimension by eliminating variables until the new equivalent X is full-dimensional in $\mathbb{R}^d$ for $d \leq n$. Furthermore, with suitable preprocessing, every hyperplane $a_j^T \mathbf{x} = b_j$ in X corresponds to a facet of X.

Let $\partial X := \{\mathbf{x} \in X \mid \exists j, a_j^T \mathbf{x} = b_j\}$ denote the boundary of X and let $\operatorname{int} X := X \setminus \partial X$ denote the interior of X. For $r \in \mathbb{R}_+$, denote the closed ball of radius r centered at the point $\mathbf{x}^0$ by $B_r(\mathbf{x}^0) = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{x}^0\|_2 \leq r\}$. A hyperplane in $\mathbb{R}^{n+1}$ through point $(\mathbf{x}^0, z^0)$ with $z^0 = f(\mathbf{x}^0)$ and normal vector $\mathbf{n}^0 \in \mathbb{R}^n, \mathbf{n}^0 \neq 0$ can be defined as

$$H(\mathbf{x}^0, \mathbf{n}^0) = \{(\mathbf{x}, z) \mid z = z^0 + \mathbf{n}^0 \cdot (\mathbf{x} - \mathbf{x}^0)\}.$$

The tangent plane to f at $(\mathbf{x}^0, f(\mathbf{x}^0))$ is $T(\mathbf{x}^0) := H(\mathbf{x}^0, \nabla f(\mathbf{x}^0))$. The downward closed halfspace associated with hyperplane $H(\mathbf{x}^0, \mathbf{n}^0)$ is then

$$\check{H}(\mathbf{x}^0, \mathbf{n}^0) = \{(\mathbf{x}, z) \mid z \geq z^0 + \mathbf{n}^0 \cdot (\mathbf{x} - \mathbf{x}^0)\}.$$

The upward closed halfspace $\hat{H}(\mathbf{x}^0, \mathbf{n}^0)$ is defined similarly.

Hyperplanes $H(\mathbf{x}^0, \mathbf{n}^0)$ are called *nonvertical* (to space $\mathbb{R}^n$), since $\pi_{\mathbf{x}}(H(\mathbf{x}^0, \mathbf{n}^0)) = \mathbb{R}^n$. Analogously, a *vertical* hyperplane in $\mathbb{R}^{n+1}$ through point $(\mathbf{x}^0, z^0)$ with $z^0 = f(\mathbf{x}^0)$ and normal vector $\mathbf{n}^0 \in \mathbb{R}^n$ can be defined as

$$H^{\perp}(\mathbf{x}^0, \mathbf{n}^0) = \{(\mathbf{x}, z) \mid \mathbf{n}^0 \cdot (\mathbf{x} - \mathbf{x}^0) = 0\}.$$

Then the left closed halfspace associated to $H^{\perp}(\mathbf{x}^0, \mathbf{n}^0)$ is defined by

$$\acute{H}^{\perp}(\mathbf{x}^0, \mathbf{n}^0) = \{(\mathbf{x}, z) \mid \mathbf{n}^0 \cdot (\mathbf{x} - \mathbf{x}^0) \leq 0\}$$

and the right closed halfspace $\grave{H}^{\perp}(\mathbf{x}^0, \mathbf{n}^0)$ is defined similarly. For any halfspace H in $\mathbb{R}^{n+1}$, let g^H denote a map which maps a halfspace to its corresponding hyperplane.

Lemma 5.2

Let $f \in \mathbb{R}[\mathbf{x}]$ be a polynomial function over X and let $\mathcal{S}$ be the graph of f as defined above. Let conv$(\mathcal{S})$ be the convex hull of $\mathcal{S}$, then it holds

$$\begin{aligned}
\mathit{conv}(\mathcal{S}) &= \left(\bigcap_{\check{H} \supset \mathcal{S}} \check{H}\right) \cap \left(\bigcap_{\hat{H} \supset \mathcal{S}} \hat{H}\right) \cap \left(\bigcap_{\acute{H}^{\perp} \supset \mathcal{S}} \acute{H}^{\perp}\right) \cap \left(\bigcap_{\grave{H}^{\perp} \supset \mathcal{S}} \grave{H}^{\perp}\right) =: S' \\
&= \left(\bigcap_{\substack{\check{H} \supset \mathcal{S} \\ g^H(\check{H}) \cap \mathcal{S} \neq \emptyset}} \check{H}\right) \cap \left(\bigcap_{\substack{\hat{H} \supset \mathcal{S} \\ g^H(\hat{H}) \cap \mathcal{S} \neq \emptyset}} \hat{H}\right) \cap \left(\bigcap_{1 \leq j \leq m} \left\{(\mathbf{x}, z) \mid a_j^T \mathbf{x} \leq b_j\right\}\right) =: S''
\end{aligned}$$

Proof. Theorem 5.1 implies that $\text{conv}(\mathcal{S}) = S'$.

We discuss first halfspaces in S' with corresponding hyperplanes which are nonvertical. For every $\check{H}^1$ there exists an affine function $g_1 : \mathbb{R}^n \to \mathbb{R}$ such that $\check{H}^1 = \{(\mathbf{x}, z) \mid z \geq g_1(\mathbf{x})\}$. Consider any $\check{H}^1$ with $\check{H}^1 \supset \mathcal{S}$, $H^1 = g^H(\check{H}^1) = \{(\mathbf{x}, z) \mid z = g_1(\mathbf{x})\}$ and $H^1 \cap \mathcal{S} = \emptyset$, i.e., the corresponding hyperplane does not support $\mathcal{S}$. It implies that $f(\mathbf{x}) > g_1(\mathbf{x})$ for all $\mathbf{x} \in X$. The function $f(\mathbf{x}) - g_1(\mathbf{x})$ attains a minimal value $d_1 \in \mathbb{R}_+$ over X since X is compact. It implies that $\check{H}^2$ denoted by $\{(\mathbf{x}, z) \mid g_1(\mathbf{x}) - d_1 \geq 0\}$ fulfills $\check{H}^2 \supset \mathcal{S}$ but $H^2 \cap \mathcal{S} \neq \emptyset$ for $H^2 = \{(\mathbf{x}, z) \mid g_1(\mathbf{x}) - d_1 = 0\}$. Note that $\check{H}^2 \subsetneq \check{H}^1$, which means to form S', every $\check{H}$ is redundant if the corresponding $H = g^H(\check{H})$ satisfies $H \cap \mathcal{S} = \emptyset$. This implies that

$$\bigcap_{\check{H} \supset \mathcal{S}} \check{H} = \bigcap_{\substack{\check{H} \supset \mathcal{S} \\ g^H(\check{H}) \cap \mathcal{S} \neq \emptyset}} \check{H}.$$

This is also true for every upward closed halfspaces.

Now we discuss halfspaces in S' with corresponding hyperplanes which are vertical. Note that for every $j \in \{1, \ldots, m\}$, $\{(\mathbf{x}, z) \mid a_j^T \mathbf{x} \leq b_j\}$ has the form of $\acute{H}^{\perp}$ with $\acute{H}^{\perp} \supset \mathcal{S}$. For any other $\acute{H}^{\perp}$ with $\acute{H}^{\perp} \supset \mathcal{S}$ or $\grave{H}^{\perp}$ with $\grave{H}^{\perp} \supset \mathcal{S}$ we have either

$$\acute{H}^{\perp} \supset \left(\bigcap_{1 \leq j \leq m} \left\{(\mathbf{x}, z) \mid a_j^T \mathbf{x} \leq b_j\right\}\right)$$

or

$$\grave{H}^{\perp} \supset \left(\bigcap_{1 \leq j \leq m} \left\{(\mathbf{x}, z) \mid a_j^T \mathbf{x} \leq b_j\right\}\right).$$

Together with the result for all halfspaces in S' with corresponding hyperplanes which are nonvertical, it implies that $S' = S''$. □

In this section, we want to find a set of halfspaces that all contain $\mathcal{S}$ and the intersection of them is the convex hull of $\mathcal{S}$. Lemma 5.2 shows that we can restrict attention to nonvertical halfspaces whose corresponding hyperplane intersects the graph.

Remark 5.3
Obviously, $\mathcal{S}$ is bounded because of Theorem 5.1. Due to symmetry, we only need to consider the downward closed part of the convex hull since the upward closed part is equivalent to the downward closed part of function $-f$.

5.3.2 Locally and globally convex points

Definition 5.4 (Locally convex points)
A point $(\mathbf{x}^0, f(\mathbf{x}^0)) \in \mathcal{S}$ is a *locally convex point* if there exists $\varepsilon > 0$, $\mathbf{n}^0 \in \mathbb{R}^n$ such that

$$\underbrace{\{(\mathbf{x}, z) \mid \mathbf{x} \in X \cap B_\varepsilon(\mathbf{x}^0), z = f(\mathbf{x})\}}_{\text{local graph}} \subset \check{H}(\mathbf{x}^0, \mathbf{n}^0). \tag{5.1}$$

In this case, the domain point $\mathbf{x}^0$ is the corresponding locally convex domain point.

Let $\check{\mathcal{S}}^l \in \mathcal{S}$ denote the set of all locally convex points and $\check{X}^l = \pi_{\mathbf{x}}(\check{\mathcal{S}}^l)$ the corresponding domain points.

Lemma 5.5
Let $\mathbf{x}^0 \in \operatorname{int} X$. If $\mathbf{x}^0$ is a locally convex domain point, then the gradient vector $\nabla f(\mathbf{x}^0)$ is the unique normal vector $\mathbf{n}^0$ to $\mathbf{x}^0$ fulfilling (5.1).

Proof. Consider the function $g_{\mathbf{n}^0}(\mathbf{x}) = f(\mathbf{x}) - (\mathbf{n}^0 \cdot (\mathbf{x} - \mathbf{x}^0) + f(\mathbf{x}^0))$ for $\mathbf{n}^0 \in \mathbb{R}^n$ and defined on X. The point $(\mathbf{x}^0, f(\mathbf{x}^0))$ is locally convex if and only if there exists $\varepsilon > 0$ such that $g_{\mathbf{n}^0}(\mathbf{x}) \geq 0$ over $X \cap B_\varepsilon(\mathbf{x}^0)$. Using "Taylor's Formula in Several Variables" for $g_{\mathbf{n}^0}(\mathbf{x})$ at $\mathbf{x}^0$ (see the book [Edw94]), $g_{\mathbf{n}^0}(\mathbf{x})$ may attain a local minimum 0 at $\mathbf{x}^0$ if and only if $\mathbf{n}^0 = \nabla f(\mathbf{x}^0)$. This implies the result. □

Definition 5.6 (Globally convex points and valid halfspaces)
A point $(\mathbf{x}^0, f(\mathbf{x}^0)) \in \mathcal{S}$ is a *globally convex point* if there exist an $\mathbf{n}^0 \in \mathbb{R}^n$ such that

$$\mathcal{S} = \underbrace{\{(\mathbf{x}, z) \mid \mathbf{x} \in X, z = f(\mathbf{x})\}}_{\text{total graph}} \subset \check{H}(\mathbf{x}^0, \mathbf{n}^0). \tag{5.2}$$

Furthermore, we call $\check{H}(\mathbf{x}^0, \mathbf{n}^0)$ a *valid* halfspace, $H(\mathbf{x}^0, \mathbf{n}^0)$ and $\mathbf{n}^0$ the corresponding valid hyperplane and valid normal vector, respectively. The point $\mathbf{x}^0$ is the corresponding globally convex domain point.

Further we define $\check{\mathcal{S}}^g$ as the set of all globally convex points and $\check{X}^g = \pi_{\mathbf{x}}(\check{\mathcal{S}}^g)$ the set of corresponding domain points.

Remark 5.7
The definition of globally convex points is actually equivalent to the definition of generating sets, which can be found in Chapter 4 of the book [LS13]. Due to the different point of view, we keep on using the name defined above in this thesis.

Theorem 5.8
Let $\mathbf{x}^0 \in \operatorname{int} X$*. Then* $\mathbf{x}^0$ *is a globally convex domain point if and only if the tangent hyperplane* $T(\mathbf{x}^0)$ *is valid.*

Proof. It is clear that a globally convex domain point $\mathbf{x}^0$ must also be a locally convex domain point. The result is then followed from Lemma 5.5, since $\mathbf{n}^0$ is the unique candidate normal vector to satisfy (5.2). □

For any $\mathbf{x}^0 \in X$ we seek a way to determine if it is a globally convex domain point. For $\mathbf{x}^0 \in \operatorname{int} X$, we need only to check if $T(\mathbf{x}^0)$ is valid due to Theorem 5.8. On the other hand, for a boundary domain point $\mathbf{x}^0 \in \partial X$, the situation is more complex. We require several additional concepts in this case.

5.3.3 Globally convex boundary points

Remark 5.9
For polytope X, every face F possesses a maximal subset $I_F \subset \{1, \dots, m\}$ such that

$$F = \left\{ \mathbf{x} \in X \mid a_j^T \mathbf{x} = b_j, j \in I_F \right\}.$$

In this section, we assume that the vertices of X are nondegenerate which means there is no point $\mathbf{x} \in \mathbb{R}^n$ which satisfies $n+1$ of the given m inequalities with equality. This implies two properties:

- For any face F it holds that $\dim F = n - |I_F|$
- For any face F with $|I_F| = d \geq 2$ and $I_F = \{1, \dots, d\}$ without loss of generality, it implies that
$$F_{-i} = \left\{ \mathbf{x} \in X \mid a_j^T \mathbf{x} = b_j, j \in I_F \setminus \{i\} \right\}$$
is also a face of X with $F \subsetneq F_{-i}$ and $\dim F = \dim F_{-i} - 1$ for any $i \in I_F$.

Note that the assumption that X is vertex-nondegenerate is not a necessary condition for this section. However, this assumption makes the notation and description easier. For a vertex-degenerate X, the dimension of any face can be determined by calculating the rank of an auxiliary matrix. Since there are finitely many faces, for any given face F with $\dim F \leq n-2$, all faces that contain F of dimension $(\dim F + 1)$ can be determined as well.

Based on this remark, we have the following lemma. Example 5.11 following after its proof illustrates the lemma in a graphical way.

Lemma 5.10 (Complete Projection based on the smallest face containing $\mathbf{x}^0$***)***
Let $\mathbf{x}^0 \in \partial X$ *and define the index set*

$$I_{\mathbf{x}^0} := \left\{ j \in \{1, \dots, m\} \mid a_j^T \mathbf{x}^0 = b_j \right\}$$

which is nonempty. Let $d' = |I_{\mathbf{x}^0}|$ *be the cardinality with* $1 \leq d' \leq n$*. Then, the set*

$$P_{\mathbf{x}^0} := \{\mathbf{x} \in \mathbb{R}^n \mid a_j^T \mathbf{x} = b_j \text{ for all } j \in I_{\mathbf{x}^0}\}$$

defines an affine set in $\mathbb{R}^n$ *of dimension* $d := n - d'$*. Further, the set* $F_{\mathbf{x}^0} := P_{\mathbf{x}^0} \cap X$ *is the smallest face of* X *which contains* $\mathbf{x}^0$ *and has dimension* d*.*

By permuting variables in $\mathbf{x}$ *if necessary, there exists a bijective linear map*

$$g_d : P_{\mathbf{x}^0} \to \mathbb{R}^d, \mathbf{x} \mapsto \mathbf{x}_d$$

where $\mathbf{x}_d = (x_1, \dots, x_d)^T$ *such that*

- *The set*
$$\mathcal{S}_d := \{(\mathbf{x}_d, z) \mid \mathbf{x}_d = g_d(\mathbf{x}), z = f(\mathbf{x}), \mathbf{x} \in F_{\mathbf{x}^0}\} \subset \mathbb{R}^{d+1}$$
is the graph of some polynomial function over a polytope.

- *Either* $(g_d(\mathbf{x}^0), f(\mathbf{x}^0))$ *is an interior point of* $\mathcal{S}_d$ *or* $\mathcal{S}_d = \{(g_d(\mathbf{x}^0), f(\mathbf{x}^0))\}$*.*

Proof. Without loss of generality, we assume $I_{\mathbf{x}^0} = \{1, \dots, d'\}$. It is clear that $F_{\mathbf{x}^0}$ is a face of X with $\mathbf{x}^0 \in F_{\mathbf{x}^0}$. Assume that there exists another face F of X with $\mathbf{x}^0 \in F \subsetneq F_{\mathbf{x}^0}$. Then there exists $j \in \{d'+1, \dots, m\}$ such that $a_j^T \mathbf{x}^0 = b_j$. This is a contradiction to the definition of $I_{\mathbf{x}^0}$. Hence $F_{\mathbf{x}^0}$ is the smallest face of X containing $\mathbf{x}^0$. Similar to the preprocessing of half spaces which form X, which shows that $F_{\mathbf{x}^0}$ has dimension d.

Furthermore, we define

$$A_{\mathbf{x}^0} = \begin{pmatrix} a_1^T \\ \vdots \\ a_{d'}^T \end{pmatrix} \in \mathbb{R}^{d' \times n},$$

and $b_{\mathbf{x}^0} = (b_1, b_2, \dots, b_{d'})^T$. Then we have

$$P_{\mathbf{x}^0} = \{\mathbf{x} \in \mathbb{R}^n \mid A_{\mathbf{x}^0}\mathbf{x} = b_{\mathbf{x}^0}\}$$

and $d' = \operatorname{rank}(A_{\mathbf{x}^0}) = n - d$. Via permuting variables, we may assume that the last d' columns of $A_{\mathbf{x}^0}$ are linearly independent and form an invertible matrix A_B. At the same time the first d columns of $A_{\mathbf{x}^0}$ form the matrix A_d. By setting $\mathbf{x}_B = (x_{d+1}, \dots, x_n)^T$, we have

$$A_{\mathbf{x}^0}\mathbf{x} = (A_d A_B)(\mathbf{x}_d^T\ \mathbf{x}_B^T)^T = A_d \mathbf{x}_d + A_B \mathbf{x}_B = b_{\mathbf{x}^0},$$

and so

$$\mathbf{x}_B = A_B^{-1}(b_{\mathbf{x}^0} - A_d\mathbf{x}_d) = A_B^{-1}b_{\mathbf{x}^0} - A_B^{-1}A_d\mathbf{x}_d =: b' - A'\mathbf{x}_d$$

for all $\mathbf{x} \in P_{\mathbf{x}^0}$ with

$$x_{d+i} = b_i' - A_i^{'}\mathbf{x}_d = b_i' + \sum_{j=1}^{d} c_{d+i,j}x_j \tag{5.3}$$

for all $i = 1, \ldots, n-d$ where $b' = A_B^{-1}b_{\mathbf{x}^0}$, $A' = A_B^{-1}A_d$ and $A_i^{'}$ is the i-th row of A', $c_{d+j,j}$ are constants for $i = 1, \ldots, n-d$ and $j = 1, \ldots, d$. Note that the d-dimensional affine set $P_{\mathbf{x}^0}$ and $\mathbb{R}^d$ are isomorphic. Consider the linear map

$$g_d : P_{\mathbf{x}^0} \to \mathbb{R}^d, \mathbf{x} \mapsto \mathbf{x}_d,$$

which is bijective since the inverse exists:

$$g_d^{-1} : \mathbb{R}^d \to P_{\mathbf{x}^0}, \mathbf{x}_d \mapsto \begin{pmatrix} \mathbf{x}_d \\ b' - A'\mathbf{x}_d \end{pmatrix}.$$

In the following we investigate the graph of function f under the restriction $\mathbf{x} \in F_{\mathbf{x}^0}$. The set $F_{\mathbf{x}^0}$, function f and point $\mathbf{x}^0$ can be *completely* projected to $F_d = g_d(F_{\mathbf{x}^0})$, $\mathbf{x}_d^0 = g_d(\mathbf{x}^0)$ and $f_d = f(\mathbf{x}_d, b_1' - A_1^{'}\mathbf{x}_d, \ldots, b_{x-d}' - A_{x-d}^{'}\mathbf{x}_d)$ respectively. It is clear that $f_d \in \mathbb{R}[\mathbf{x}_d]$ is a polynomial function and F_d is a polytope of dimension d. By comparing coefficients, for every $\mathbf{x} \in F_{\mathbf{x}^0}$ it implies that $f(\mathbf{x}) = f_d(g_d(\mathbf{x}))$ and for every $\mathbf{x}_d \in F_d$ it implies that $f_d(\mathbf{x}_d) = f(g_d^{-1}(\mathbf{x}_d))$. The graph of f_d is thus

$$\{(\mathbf{x}_d, z) \mid z = f_d(\mathbf{x}_d),\ \mathbf{x}_d \in F_d\} = \{(\mathbf{x}_d, z) \mid \mathbf{x}_d = g_d(\mathbf{x}), z = f(\mathbf{x}), \mathbf{x} \in F_{\mathbf{x}^0}\} = \mathcal{S}_d,$$

which is the graph of the polynomial function f_d over polytope F_d.

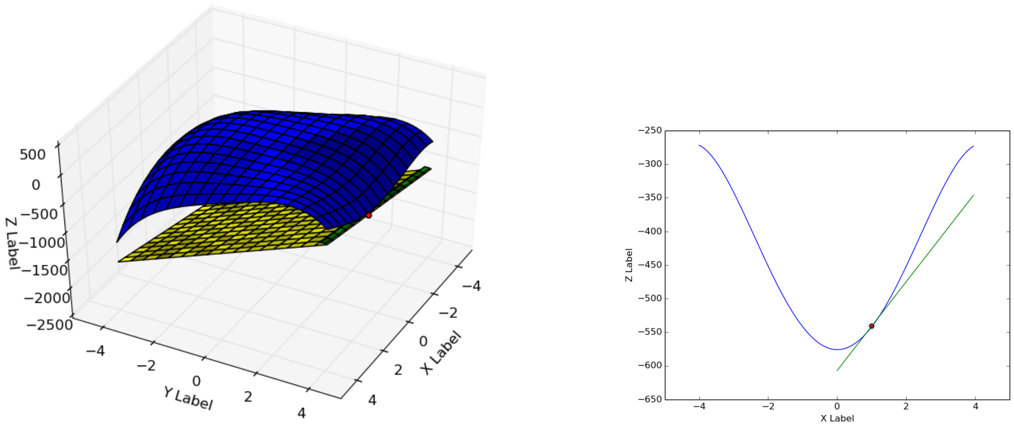
The domain point $\mathbf{x}^0$ is an extreme point of X if $d' = |I_{\mathbf{x}^0}| = n$. In this case $d = 0$ and $F_d = \{\mathbf{x}_d^0\}$ which contains a single point. It implies that $\mathbf{x}_d^0$ is also an extreme point of F_d. Otherwise, we are in case of $d' < n$. Note that every F_d is isomorphic to a face of X. Assume $\mathbf{x}_d^0$ is not an interior point of F_d. Then there exists a face of F_d which contains $\mathbf{x}_d^0$. It implies there exists a face of X which contains $\mathbf{x}^0$. This is a contradiction to the definition of $I_{\mathbf{x}^0}$. Thus $\mathbf{x}_d^0$ is an interior point in F_d. □

Example 5.11
We look at an example with $n = 2$, i.e., $\mathbf{x} = (x, y)$. Consider the polynomial function

$$f(x, y) = -x^4 + 3x^3 + 10x^2y - 5x^2 + 3x - 2y^4 - 4y^2 : [-4, 4] \times [-4, 4] \to \mathbb{R}.$$

For boundary point $\mathbf{x}^0 = (1, 4)$ we get $F_{\mathbf{x}^0} = \{(x, y) \mid -4 \le x \le 4, y = 4\}$, $g_d : (x, y) \mapsto x$ and $g_d^{-1} : x \mapsto (x, 4)$ and $f_d(x) = -x^4 + 3x^3 + 35x^2 + 3x - 576$ as defined in Lemma 5.10.

Point $(\mathbf{x}^0, f(\mathbf{x}^0))$, set $\mathcal{S}$ and a valid hyperplane $\{(x, y, z) \mid z = 78x + 134y - 1150\}$ are shown as the red point, the blue surface and the yellow plane in Figure 5.4a and the corresponding $(\mathbf{x}_d^0, f_d(\mathbf{x}_d^0))$ and $\mathcal{S}_d$ are shown as the red point and the blue curve in Figure 5.4b, respectively.♢



(a) In space $\mathbb{R}^{n+1}$

(b) In space $\mathbb{R}^{d+1}$

Figure 5.4: Graph and hyperplane restricted on a face in Example 5.11 and the complete projections

Theorem 5.12 (Globally convex boundary points)
Let $\mathbf{x}^0 \in \partial X$. Then $\mathbf{x}^0$ is a globally convex domain point of $\mathcal{S}$ if and only if $\mathbf{x}_d^0 = g_d(\mathbf{x}^0)$ is also a globally convex domain point of $\mathcal{S}_d$ with

$$\mathcal{S}_d = \{(\mathbf{x}_d, z) \mid \mathbf{x}_d = g_d(\mathbf{x}), z = f(\mathbf{x}), \mathbf{x} \in F_{\mathbf{x}^0}\} \subset \mathbb{R}^{d+1}.$$

We go back to Example 5.11. Point $\mathbf{x}_d^0 = 1$ is a globally convex domain point of $\mathcal{S}_d$ since the tangent plane $\{(x, z) \mid z = f_d(x) = 78x - 614\}$ (shown as green line in Figure 5.4b) can be verified to be valid. Using the theorem above it implies that $(1, 4)$ is also globally convex. This is verified by the valid hyperplane $\{(x, y, z) \mid z = 78x + 134y - 1150\}$.

We prove first the direction "⇒" and postpone the direction "⇐" after introducing a few further definitions and auxiliary lemmas.

Proof (of "⇒"). Consider the case that $\mathbf{x}_d^0$ and $\mathbf{x}^0$ are both an extreme point in F_d and X, respectively. Then the single point $\mathbf{x}_d^0 \in F_d$ is surely a globally convex domain point since the single point is a valid hyperplane in that space.

Otherwise $\mathbf{x}_d^0$ is an interior point in F_d. Let $\mathbf{v} = (v_1, \ldots, v_n)^T \in \mathbb{R}^n$ be an arbitrary vector.

Define a function $g_c : \mathbb{R}^n \to \mathbb{R}^d$,

$$\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \\ v_{d+1} \\ \vdots \\ v_n \end{pmatrix} \mapsto \begin{pmatrix} v_1 + \sum_{i=d+1}^n c_{i,1} v_i \\ v_2 + \sum_{i=d+1}^n c_{i,2} v_i \\ \vdots \\ v_d + \sum_{i=d+1}^n c_{i,d} v_i \end{pmatrix},$$

where all c_{ij} are defined as in (5.3). We first give and prove the following claim.

Claim Let $\mathbf{n} \in \mathbb{R}^n$. The following two sets are affinely isomorphic

$$\{(\mathbf{x}, z) \mid z = z^0 + \mathbf{n} \cdot (\mathbf{x} - \mathbf{x}^0), \mathbf{x} \in P_{\mathbf{x}^0}\} \cong \{(\mathbf{x}_d, z) \mid z = z^0 + g_c(\mathbf{n}) \cdot (\mathbf{x}_d - \mathbf{x}_d^0), \mathbf{x}_d \in \mathbb{R}^d\}. \quad (5.4)$$

This implies

$$\{(\mathbf{x}, z) \mid z \geq z^0 + \mathbf{n} \cdot (\mathbf{x} - \mathbf{x}^0), \mathbf{x} \in P_{\mathbf{x}^0}\} \cong \{(\mathbf{x}_d, z) \mid z \geq z^0 + g_c(\mathbf{n}) \cdot (\mathbf{x}_d - \mathbf{x}_d^0), \mathbf{x}_d \in \mathbb{R}^d\}. \quad (5.5)$$

Moreover, the tangent plane on $\mathbf{x}_d^0$ defined as

$$T_d(\mathbf{x}_d^0) = \{(\mathbf{x}_d, z) \mid z = \nabla f_d(\mathbf{x}_d^0) \cdot (\mathbf{x}_d - \mathbf{x}_d^0) + z^0, \mathbf{x}_d \in \mathbb{R}^d\}$$

satisfies

$$T_d(\mathbf{x}_d^0) \cong T_{|P_{\mathbf{x}^0}}(\mathbf{x}^0) = \{(\mathbf{x}, z) \in T(\mathbf{x}^0) \mid \mathbf{x} \in P_{\mathbf{x}^0}\}. \quad (5.6)$$

We prove the above claim first. For any $\mathbf{n} = (\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_n)^T \in \mathbb{R}^n$,

$$\begin{aligned}
\mathbf{n} \cdot (\mathbf{x} - \mathbf{x}^0) + z^0 &= \sum_{i=1}^{n} \mathbf{n}_i (x_i - x_i^0) + z^0 \\
&= \sum_{i=1}^{d} \mathbf{n}_i (x_i - x_i^0) + \sum_{i=d+1}^{n} \mathbf{n}_i (x_i - x_i^0) + z^0 \\
&\overset{(5.3)}{=} \sum_{i=1}^{d} \mathbf{n}_i (x_i - x_i^0) + \sum_{i=d+1}^{n} \mathbf{n}_i \left(b'_{i-d} + \sum_{j=1}^{d} c_{i,j} x_j - b'_{i-d} - \sum_{j=1}^{d} c_{i,j} x_j^0 \right) + z^0 \\
&= \sum_{i=1}^{d} \mathbf{n}_i (x_i - x_i^0) + \underbrace{\sum_{i=d+1}^{n} \mathbf{n}_i \left(\sum_{j=1}^{d} c_{i,j} (x_j - x_j^0) \right)}_{\text{change } i \text{ and } j} + z^0 \\
&= \sum_{i=1}^{d} \mathbf{n}_i (x_i - x_i^0) + \sum_{j=d+1}^{n} \mathbf{n}_j \left(\sum_{i=1}^{d} c_{j,i} (x_i - x_i^0) \right) + z^0 \\
&= \sum_{i=1}^{d} \mathbf{n}_i (x_i - x_i^0) + \sum_{i=1}^{d} \sum_{j=d+1}^{n} c_{j,i} \mathbf{n}_j (x_i - x_i^0) + z^0 \\
&= \sum_{i=1}^{d} \left(\mathbf{n}_i + \sum_{j=d+1}^{n} c_{j,i} \mathbf{n}_j \right) (x_i - x_i^0) + z^0 \\
&= g_c(\mathbf{n}) \cdot (\mathbf{x}_d - \mathbf{x}_d^0) + z^0
\end{aligned}$$

for all $\mathbf{x} \in P_{\mathbf{x}^0}, \mathbf{x}_d \in \mathbb{R}^d$ with $\mathbf{x}_d = g_d(\mathbf{x})$. This implies (5.4) and (5.5) holds analogously.

To prove (5.6), it suffices to show

$$g_c(\nabla f(\mathbf{x}^0)) = \nabla f_d(\mathbf{x}_d^0),$$

which is true if for every x_j with $j = 1 \dots, d$ we have

$$\frac{\partial f_d}{\partial x_j}(\mathbf{x}_d^0) = \frac{\partial f}{\partial x_j}(\mathbf{x}^0) + \sum_{i=d+1}^{n} c_{ij} \frac{\partial f}{\partial x_i}(\mathbf{x}^0). \tag{5.7}$$

Since

$$f_d(\mathbf{x}_d) = f(x_1, x_2, \dots, x_d, b'_1 + \sum_{j=1}^{d} c_{d+1,j} x_j, \dots, b'_{n-d} + \sum_{j=1}^{d} c_{n,j} x_j),$$

we get (5.7) by using the chain rule when computing $\frac{\partial f_d}{\partial x_j}(\mathbf{x}_d^0)$.

Equation (5.4) says that for a given hyperplane $H(\mathbf{x}^0, \mathbf{n})$, the graph with restriction $\mathbf{x} \in P_{\mathbf{x}^0}$ in $\mathbb{R}^{n+1}$ which is $\{(\mathbf{x}, z) \mid z = \mathbf{n}(\mathbf{x} - \mathbf{x}^0) - z^0, \mathbf{x} \in P_{\mathbf{x}^0}\}$ is affinely isomorphic to the hyperplane

$H(\mathbf{x}_d^0, g_c(\mathbf{n}))$ in $\mathbb{R}^{d+1}$. Equation (5.5) translates (5.4) to halfspaces, and (5.6) similarly for tangent hyperplanes.

Now we are ready to prove the remaining part of "$\Rightarrow$". Let $H = \{(\mathbf{x}, z) \mid z = z^0 + \mathbf{n}^0 \cdot (\mathbf{x} - \mathbf{x}^0)\}$ be the hypothesized hyperplane for showing globally convexity. Similar to the proof of the claim above, it implies that

$$\underbrace{\{\mathbf{x} \in F_{\mathbf{x}^0} \mid f(\mathbf{x}) \geq \mathbf{n} \cdot (\mathbf{x} - \mathbf{x}^0) + z^0\}}_{=:L_1} \cong \underbrace{\{\mathbf{x}_d \in F_d \mid f_d(\mathbf{x}_d) \geq g_c(\mathbf{n}) \cdot (\mathbf{x}_d - \mathbf{x}_d^0) + z^0\}}_{=:R_1}.$$

Note that if H is valid, then we have $L_1 = F_{\mathbf{x}^0}$, which is equivalent to the case that $T_{I_{\mathbf{x}^0}}(\mathbf{x}^0)$ is below $\mathcal{S}$ over $F_{\mathbf{x}^0}$ and is isomorphic to $R_1 = F_d$. This implies that $\mathbf{x}_d^0$ is a globally convex domain point in $\mathbb{R}^d$. □

Corollary 5.13
Let $\mathbf{x}^0 \in \partial X$ with corresponding $I_{\mathbf{x}^0}$. Define $T_{I_{\mathbf{x}^0}}(\mathbf{x}^0) = T(\mathbf{x}^0) \cap \{(\mathbf{x}, z) \mid a_i^T \mathbf{x} = b_i, i \in I_{\mathbf{x}^0}\}$. Every valid hyperplane H through $(\mathbf{x}^0, f(\mathbf{x}^0))$ contains the affine set $T_{I_{\mathbf{x}^0}}(\mathbf{x}^0)$, i.e.,

$$T_{I_{\mathbf{x}^0}}(\mathbf{x}^0) \subset H.$$

Proof. We use all definitions and results from the proof of "$\Rightarrow$" of Theorem 5.12.

If $\mathbf{x}_d^0$ and $\mathbf{x}^0$ are both an extreme point in F_d and X respectively, then we have

$$T_{I_{\mathbf{x}^0}}(\mathbf{x}^0) = \{(\mathbf{x}^0, f(\mathbf{x}^0))\} \subset H.$$

Otherwise $\mathbf{x}_d^0$ is an interior point in F_d. Similar to the proof of Equation (5.4) in the proof of "$\Rightarrow$" of Theorem 5.12, it implies for $H = H(\mathbf{x}^0, \mathbf{n})$ that

$$\underbrace{\{(\mathbf{x}, z) \mid z = \mathbf{n} \cdot (\mathbf{x} - \mathbf{x}^0) + z^0, \mathbf{x} \in P_{\mathbf{x}^0}\}}_{=:L_2} \cong \underbrace{\{(\mathbf{x}_d, z) \mid z = g_c(\mathbf{n}) \cdot (\mathbf{x}_d - \mathbf{x}_d^0) + z^0, \mathbf{x}_d \in \mathbb{R}^d\}}_{=:R_2}.$$

Note that $L_2 = H \cap \{(\mathbf{x}, z) \mid \mathbf{x} \in P_{\mathbf{x}^0}\}$ and

$$\begin{aligned} H \text{ is valid } &\Rightarrow L_1 = F_{\mathbf{x}^0} \\ &\Leftrightarrow R_1 = F_d \\ &\overset{(*)}{\Leftrightarrow} g_c(\mathbf{n}) = \nabla f(\mathbf{x}_d^0) \\ &\Leftrightarrow R_2 = T_d(\mathbf{x}_d^0) \\ &\Leftrightarrow L_2 = T_{I_{\mathbf{x}^0}}(\mathbf{x}^0) \end{aligned}$$

and thus

$$T_{I_{\mathbf{x}^0}}(\mathbf{x}^0) \subset H,$$

where $(*)$ is followed from Theorem 5.8. □

Recall Example 5.11 again. We have $T_{I_{\mathbf{x}^0}}(\mathbf{x}^0) = \{(x, y, z) \mid z = 78x - 614, y = 4\}$ for the globally convex point $\mathbf{x}^0 = (1, 4)$, see the green line in Figure 5.4. Corollary 5.13 shows that every valid hyperplane through $(\mathbf{x}^0, f(\mathbf{x}^0))$ contains $T_{I_{\mathbf{x}^0}}(\mathbf{x}^0)$. Note that $T_{I_{\mathbf{x}^0}}(\mathbf{x}^0)$ is an affine set contained in the tangent plane $T(\mathbf{x}^0)$, we call it a subtangent plane. We give a general definition of subtangent planes that are contained in a valid hyperplane.

Definition 5.14 (Subtangent planes)
Let $\mathbf{x}^0 \in \partial X$ with corresponding $I_{\mathbf{x}^0}$. For every $I \subset I_{\mathbf{x}^0}$, a corresponding *subtangent plane* $T_I(\mathbf{x}^0)$ is defined as

$$T_I(\mathbf{x}^0) = T(\mathbf{x}^0) \cap \{(\mathbf{x}, z) \mid a_i^T \mathbf{x} = b_i, i \in I\}.$$

A subtangent plane $T_I(\mathbf{x}^0)$ is *valid* if there exists a valid hyperplane H with $T_I(\mathbf{x}^0) \subset H$. Note that $T(\mathbf{x}^0)$ may not be valid and yet $T_I(\mathbf{x}^0)$ is valid by choosing $H \neq T(\mathbf{x}^0)$. A valid subtangent plane through $(\mathbf{x}^0, f(\mathbf{x}^0))$ is said to be *maximally valid* if there does not exist any other valid subtangent plane $T_{I'}(\mathbf{x}^0)$ such that $T_I(\mathbf{x}^0) \subsetneq T_{I'}(\mathbf{x}^0)$.

We also extend the definition to $\mathbf{x}^0 \in \operatorname{int} X$ since $I_{\mathbf{x}^0} = \emptyset$ and $T_\emptyset(\mathbf{x}^0) = T(\mathbf{x}^0)$.

Note that since $|I_{\mathbf{x}^0}| = d'$ there are $2^{d'}$ such subtangent planes and all of them are affine sets. For every I_1, I_2 with $I_1, I_2 \subset I_{\mathbf{x}^0}$ it implies that

$$\operatorname{aff}\{T_{I_1}(\mathbf{x}^0), T_{I_2}(\mathbf{x}^0)\} = T_{I_1 \cap I_2}(\mathbf{x}^0)$$

and it is clear that

$$T_{I_1}(\mathbf{x}^0) \supset T_{I_2}(\mathbf{x}^0)$$

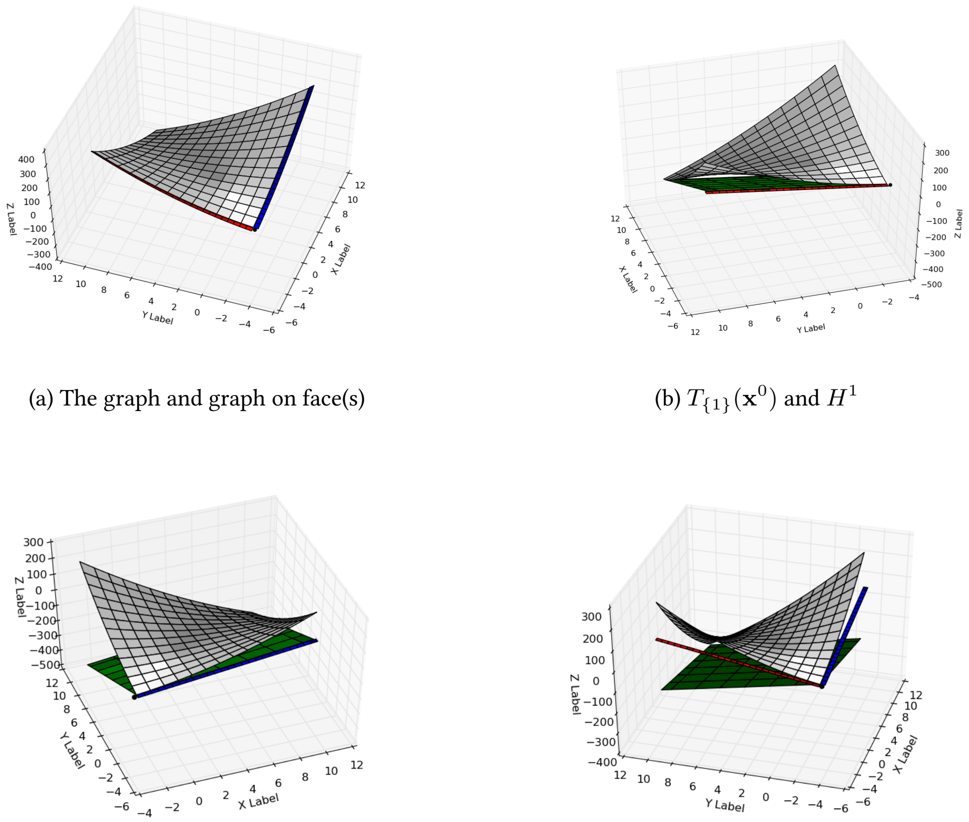
for $I_1 \subset I_2 \subset I_{\mathbf{x}^0}$.

Corollary 5.13 shows that every valid hyperplane H through a boundary point $(\mathbf{x}^0, f(\mathbf{x}^0))$ satisfies

$$H \supset T_{I_{\mathbf{x}^0}}(\mathbf{x}^0) = \Big(T(\mathbf{x}^0) \cap \{(\mathbf{x}, z) \mid \mathbf{x} \in P_{\mathbf{x}^0}\}\Big) \supset \Big(T(\mathbf{x}^0) \cap \{(\mathbf{x}, z) \mid \mathbf{x} \in F_{\mathbf{x}^0}\}\Big).$$

Face $F_{\mathbf{x}^0}$ of dimension less than $n - 1$ is contained in a face of X of higher dimension. Example 5.15 below shows that there could exist another face $F^1_{\mathbf{x}^0}$ of X such that $F^1_{\mathbf{x}^0} \supsetneq F_{\mathbf{x}^0}$ and there exists also a valid hyperplane H^1 with $H^1 \supset (T(\mathbf{x}^0) \cap \{(\mathbf{x}, z) \mid \mathbf{x} \in F^1_{\mathbf{x}^0}\})$.

Example 5.15
Consider the following example in $\mathbb{R}^3$. Let $f(x, y) = x^2 - 5xy + y^2$ be the polynomial function over domain $X = \{(x, y) \in [-3, 10] \times [-3, 10]\}$, shown in Figure 5.6. The inequalities for the halfspaces of X with $x \geq -3$, $y \geq -3$, $x \leq 10$ and $y \leq 10$ have corresponding index $1, 2, 3$ and 4, respectively. The graph of f in $\mathbb{R}^3$ is illustrated in Figure 5.5a. The domain point $\mathbf{x}^0 = (-3, -3) \in X$ is a boundary point with $I_{\mathbf{x}^0} = \{1, 2\}$. Let $F_1 = \{(x, y) \mid x = -3, -3 \leq y \leq 10\}$ and $F_2 = \{(x, y) \mid -3 \leq x \leq 10, y = -3\}$ be two faces of X containing $\mathbf{x}^0$. The



(a) The graph and graph on face(s)

(b) $T_{\{1\}}(\mathbf{x}^0)$ and H^1

(c) $T_{\{2\}}(\mathbf{x}^0)$ and H^2

(d) $T_{\{1,2\}}(\mathbf{x}^0)$ and H^3

Figure 5.5: Example 5.15 for valid subtangent planes and corresponding valid hyperplanes

graph of f with restriction $\mathbf{x} \in F_1$ is the red curve in Figure 5.5a and the graph of f with restriction $\mathbf{x} \in F_2$ is the blue curve in Figure 5.5a. In addition, the $2^2 = 4$ subtangent planes are

$$\begin{aligned}
T_{\emptyset}(\mathbf{x}^0) &= \{(x, y, z) \mid z = 9x + 9y + 27\} = T(\mathbf{x}^0),\\
T_{\{1\}}(\mathbf{x}^0) &= T(\mathbf{x}^0) \cap \{(x, y, z) \mid x = -3\} = \{(x, y, z) \mid z = 9y, x = -3\},\\
T_{\{2\}}(\mathbf{x}^0) &= T(\mathbf{x}^0) \cap \{(x, y, z) \mid y = -3\} = \{(x, y, z) \mid z = 9x, y = -3\},\\
T_{\{1,2\}}(\mathbf{x}^0) &= T(\mathbf{x}^0) \cap \{(x, y, z) \mid x = -3, y = -3\} = \{(-3, -3, -27)\}.
\end{aligned}$$

Consider the point $\mathbf{x}^1 = (10, 10)$ with $f(\mathbf{x}^1) = -300$. In the following, we investigate whether $T_{\emptyset}(\mathbf{x}^0)$ is valid. It is clear that the point $(10, 10, 9 \cdot 10 + 9 \cdot 10 + 27) = (10, 10, 207) \in T_{\emptyset}(\mathbf{x}^0)$. Note that the point $P^1 = (\mathbf{x}^1, f(\mathbf{x}^1)) = (10, 10, -300)$ is below $T_{\emptyset}(\mathbf{x}^0)$ because P^1 is below the point $(10, 10, 207)$. Hence $T_{\emptyset}(\mathbf{x}^0)$ is not valid. The subtangent plane $T_{\{1\}}(\mathbf{x}^0)$ (red line

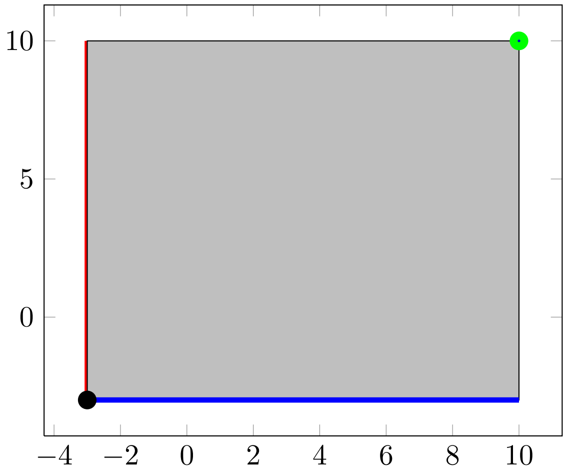


Figure 5.6: Domain $X = \{(x, y) \in [-3, 10] \times [-3, 10]\}$

in Figure 5.5b and in Figure 5.5d) is valid since the hyperplane H^1 (green plane in Figure 5.5b) with

$$H^1 = \operatorname{aff}\{T_{\{1\}}, \{P^1\}\} = \{(x, y, z) \mid z = -30x + 9y - 90\}$$

can be verified to be valid and $T_{\{1\}}(\mathbf{x}^0) \subset H^1$. Similarly, $T_{\{2\}}(\mathbf{x}^0)$ (blue line in Figure 5.5c and in Figure 5.5d) is valid since the hyperplane H^2 (green plane in Figure 5.5c) with

$$H^2 = \operatorname{aff}\{T_{\{2\}}, \{P^1\}\} = \{(x, y, z) \mid z = 9x - 30y - 90\}$$

can be verified to be valid and $T_{\{2\}}(\mathbf{x}^0) \subset H^2$. Let

$$H^3 = \{(x, y, z) \mid z = -10.5x - 10.5y - 90\}$$

with $\{(-3, -3, -27)\} \subset H^3$ and $\{P^1\} \subset H^3$. The affine set H^3 (green plane in Figure 5.5d) can also be verified to be valid. The subtangent plane $T_{\{1,2\}}(\mathbf{x}^0)$ (black point in Figure 5.5d) is valid since $T_{\{1,2\}}(\mathbf{x}^0) \subset H^3$.

Note that $T_{\{1\}}(\mathbf{x}^0)$ and $T_{\{2\}}(\mathbf{x}^0)$, both through $(\mathbf{x}^0, f(\mathbf{x}^0))$, are maximally valid subtangent planes. Hence a maximally valid subtangent plane does not have to be unique. $\diamondsuit$

Definition 5.16

Let $S_1, S_2 \subset \mathbb{R}^{n+1}$ be two sets. For a given domain $D \subset \mathbb{R}^n$, S_1 is said to be *(strictly) below* S_2 over D if $\pi_{\mathbf{x}}(S_1) \cap \pi_{\mathbf{x}}(S_2) \cap D \neq \emptyset$ and for every $\mathbf{x}^0 \in \pi_{\mathbf{x}}(S_1) \cap \pi_{\mathbf{x}}(S_2) \cap D$ and $z^1, z^2 \in \mathbb{R}$ with $(\mathbf{x}^0, z^1) \in S_1$ and $(\mathbf{x}^0, z^2) \in S_2$, we have

$$z^1 \leq z^2 (z^1 < z^2).$$

In this case we also call S_2 is *(strictly) above* S_1 over D. We omit "over D" if

$$D \supset (\pi_{\mathbf{x}}(S_1) \cap \pi_{\mathbf{x}}(S_2)) .$$

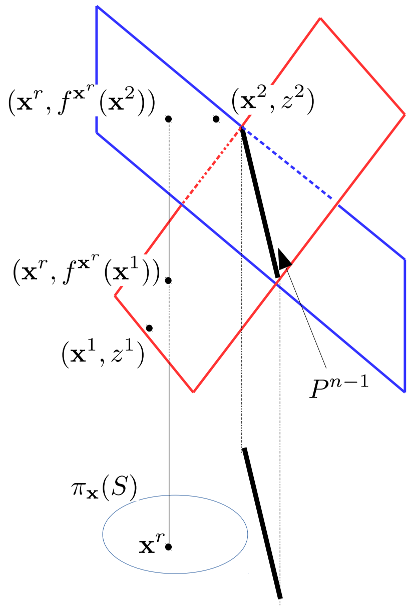

$(\mathbf{x}^r, f^{\mathbf{x}^r}(\mathbf{x}^2))$
$(\mathbf{x}^2, z^2)$
$(\mathbf{x}^r, f^{\mathbf{x}^r}(\mathbf{x}^1))$
$(\mathbf{x}^1, z^1)$
P^{n-1}
$\pi_{\mathbf{x}}(S)$
$\mathbf{x}^r$

Figure 5.7: Example of half hyperplanes for Lemma 5.18 and Lemma 5.19

Remark 5.17
In the definition above, the comparison *below* or *above* allows the compared sets to have intersection points. Note that for the special case $S_1 = S_2$, we also have S_1 is above S_2 as well as S_1 is below S_2.

Note that a hyperplane H is valid if and only if it is below $\mathcal{S}$. In the example shown in Figure 5.11a, H^i is below H^j over $[l, u]$ for all i, j with $1 \leq i < j \leq 4$ and H^i is below $\mathcal{S}$ for all $i \in \{1, 2, 3, 4\}$. Note that for any $S_1, S_2, S_3 \subset \mathbb{R}^{n+1}$ with S_1 is below S_2 over $D^1 \subset \mathbb{R}^n$ and S_2 is below S_3 over $D^2 \subset \mathbb{R}^n$, S_1 is below S_3 over $D^1 \cap D^2$.

Lemma 5.18 (Half hyperplanes)
Let

$$H^1 = H(\mathbf{x}^0, \mathbf{n}^1) = \{(\mathbf{x}, z) \mid z = z^0 + \mathbf{n}^1 \cdot (\mathbf{x} - \mathbf{x}^0)\}$$

with the normal vector $\mathbf{n}^1 \in \mathbb{R}^n$ *and*

$$H^2 = H(\mathbf{x}^0, \mathbf{n}^2) = \{(\mathbf{x}, z) \mid z = z^0 + \mathbf{n}^2 \cdot (\mathbf{x} - \mathbf{x}^0)\}$$

with the normal vector $\mathbf{n}^2 \in \mathbb{R}^n$ *be two nonvertical and nonparallel hyperplanes with an intersection point* $(\mathbf{x}^0, z^0) \in H^1 \cap H^2$. *Let*

$$\pi_{\mathbf{x}}(H^1 \cap H^2) = \{\mathbf{x} \in \mathbb{R}^n | z^0 + \mathbf{n}^1 \cdot (\mathbf{x} - \mathbf{x}^0) = z^0 + \mathbf{n}^2 \cdot (\mathbf{x} - \mathbf{x}^0)\} = \{\mathbf{x} \in \mathbb{R}^n | (\mathbf{n}^1 - \mathbf{n}^2) \cdot (\mathbf{x} - \mathbf{x}^0) = 0\}.$$

Then H^1 is below H^2 over $L := \{\mathbf{x} \in \mathbb{R}^n | (\mathbf{n}^1 - \mathbf{n}^2) \cdot (\mathbf{x} - \mathbf{x}^0) \leq 0\}$.

In addition, for any set S with $S \subset L$ or $S \subset R := \{\mathbf{x} \in \mathbb{R}^n | (\mathbf{n}^1 - \mathbf{n}^2) \cdot (\mathbf{x} - \mathbf{x}^0) \geq 0\}$, either H^1 is below H^2 over S or H^2 is below H^1 over S.

Proof. For any $\mathbf{x} \in L$ we compare $(\mathbf{x}, z^1) \in H^1$ and $(\mathbf{x}, z^2) \in H^2$ with

$$z^1 - z^2 = (z^0 + \mathbf{n}^1 \cdot (\mathbf{x} - \mathbf{x}^0)) - (z^0 + \mathbf{n}^2 \cdot (\mathbf{x} - \mathbf{x}^0)) = (\mathbf{n}^1 - \mathbf{n}^2) \cdot (\mathbf{x} - \mathbf{x}^0) \leq 0,$$

which means H^1 is below H^2 over L.

After that, the second part is trivial to prove. □

Note that if we write $\pi_{\mathbf{x}}(H^1 \cap H^2) = \{\mathbf{x} \in \mathbb{R}^n | (\mathbf{n}^2 - \mathbf{n}^1) \cdot (\mathbf{x} - \mathbf{x}^0) = 0\}$, then H^1 is above H^2 over $\{\mathbf{x} \in \mathbb{R}^n | (\mathbf{n}^2 - \mathbf{n}^1) \cdot (\mathbf{x} - \mathbf{x}^0) \leq 0\} = \{\mathbf{x} \in \mathbb{R}^n | (\mathbf{n}^1 - \mathbf{n}^2) \cdot (\mathbf{x} - \mathbf{x}^0) \geq 0\} = \mathbb{R}$. An example is shown in Figure 5.7.

Lemma 5.19

Consider the set $P^{n-1} = \{(\mathbf{x}, z) \mid z = z^0 + \mathbf{n}^1 \cdot (\mathbf{x} - \mathbf{x}^0), (\mathbf{n}^1 - \mathbf{n}^2) \cdot (\mathbf{x} - \mathbf{x}^0) = 0\}$ for $\mathbf{n}^1, \mathbf{n}^2 \in \mathbb{R}^n$, $\mathbf{n}^1 \neq \mathbf{n}^2$ and $\mathbf{x}^0 \in X, z^0 \in \mathbb{R}$. Then P^{n-1} is an affine set of dimension $(n-1)$. For any compact set $S \in \mathbb{R}^{n+1}$ with $\pi_{\mathbf{x}}(S) \subset \{\mathbf{x} \mid (\mathbf{n}^1 - \mathbf{n}^2) \cdot (\mathbf{x} - \mathbf{x}^0) < 0\}$ or $\pi_{\mathbf{x}}(S) \subset \{\mathbf{x} \mid (\mathbf{n}^1 - \mathbf{n}^2) \cdot (\mathbf{x} - \mathbf{x}^0) > 0\}$, there exists a hyperplane $H \supset P^{n-1}$ which is below S.

Proof. The set P^{n-1} can be equivalently written as

$$P^{n-1} = \{(\mathbf{x}, z) \mid z = z^0 + \mathbf{n}^1 \cdot (\mathbf{x} - \mathbf{x}^0)\} \cap \{(\mathbf{x}, z) \mid z = z^0 + \mathbf{n}^2 \cdot (\mathbf{x} - \mathbf{x}^0)\},$$

which is an intersection of two nonparallel hyperplanes. Hence it is an affine set of dimension $(n-1)$.

Note that for any $(\mathbf{x}, z) \in S$, we have $\mathbf{x} \notin \pi_{\mathbf{x}}(P^{n-1})$. The affine set $\operatorname{aff}\{P^{n-1}, \{(\mathbf{x}, z)\}\}$ is then a nonvertical hyperplane. We take an arbitrary fixed point $\mathbf{x}^r \in \pi_{\mathbf{x}}(S)$ as a *reference point*, with an affine set $\{(\mathbf{x}^r, z) \mid z \in \mathbb{R}\}$. For any $(\mathbf{x}, z) \in S$, $\operatorname{aff}\{P^{n-1}, \{(\mathbf{x}, z)\}\}$ has exactly one intersecting point with the affine set $\{(\mathbf{x}^r, z) \mid z \in \mathbb{R}\}$, see examples in Figure 5.7 with $(\mathbf{x}, z) = (\mathbf{x}^1, z^1)$ or $(\mathbf{x}, z) = (\mathbf{x}^2, z^2)$. Denoting the intersection point by $(\mathbf{x}^r, f^{\mathbf{x}^r}(\mathbf{x}))$ with

$$\{(\mathbf{x}^r, f^{\mathbf{x}^r}(\mathbf{x}))\} = \operatorname{aff}\{P^{n-1}, \{(\mathbf{x}, z) \in S\}\} \cap \{(\mathbf{x}^r, z) \mid z \in \mathbb{R}\}$$

and $f^{\mathbf{x}^r} : \pi_{\mathbf{x}}(S) \to \mathbb{R}$. Function $f^{\mathbf{x}^r}$ is continuous since the affine function above is continuous.

Let $(\mathbf{x}^1, z^1) \in S$ and $(\mathbf{x}^2, z^2) \in S$. Then the hyperplane $\operatorname{aff}\{P^{n-1}, \{(\mathbf{x}^1, z^1)\}\}$ is below $\operatorname{aff}\{P^{n-1}, \{(\mathbf{x}^2, z^2)\}\}$ over S if and only if $f^{\mathbf{x}^r}(\mathbf{x}^1) \leq f^{\mathbf{x}^r}(\mathbf{x}^2)$, see an example in Figure 5.7.

Due to the Weierstrass Theorem, we can find $(\mathbf{x}^*, z^*) \in S$ such that $f^{\mathbf{x}^r}$ attains a minimum at $\mathbf{x}^*$. Then we have $\operatorname{aff}\{P^{n-1}, \{(\mathbf{x}^*, z^*)\}\}$ is below $\operatorname{aff}\{P^{n-1}, \{(\mathbf{x}^i, z^i)\}\}$ for any $(\mathbf{x}^i, z^i) \in S$. This implies that $\operatorname{aff}\{P^{n-1}, \{(\mathbf{x}^*, z^*)\}\}$ is below any $(\mathbf{x}^i, z^i) \in S$. Hence the hyperplane $H := \operatorname{aff}\{P^{n-1}, \{(\mathbf{x}^*, z^*)\}\}$ is below S. □

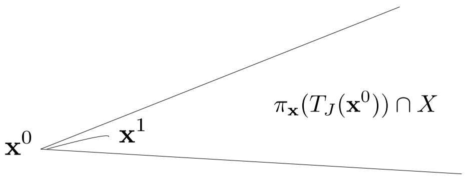


Figure 5.8: A point sequence in the neighborhood of $\mathbf{x}^0$

With the following lemma we prove that a hyperplane H is valid if and only if H is below $\mathcal{S}$ over $\check{X}^g$.

Lemma 5.20
The hyperplane $H(\mathbf{x}^0, \mathbf{n}^0)$ is below $\mathcal{S}$ over X if and only if $H(\mathbf{x}^0, \mathbf{n}^0)$ is below $\mathcal{S}$ over $\check{X}^g$, which means for every $\mathbf{x} \in \check{X}^g$ it satisfies

$$f(\mathbf{x}) \geq f(\mathbf{x}^0) + \mathbf{n}^0 \cdot (\mathbf{x} - \mathbf{x}^0). \tag{5.8}$$

Proof. Direction "$\Rightarrow$“ simply follows since $X \supset \check{X}^g$.

For the proof of "$\Leftarrow$“, first consider the optimization problem

$$\min_{\mathbf{x} \in X} \quad f(\mathbf{x}) - f(\mathbf{x}^0) - \mathbf{n}^0 \cdot (\mathbf{x} - \mathbf{x}^0), \tag{5.9}$$

which has a continuous objective function over a compact feasible region. Since X is compact, an optimum always exists due to the Weierstrass Theorem. Due to $\mathbf{x}^0 \in X$, we have

$$\min_{\mathbf{x} \in X} \quad f(\mathbf{x}) - f(\mathbf{x}^0) - \mathbf{n}^0 \cdot (\mathbf{x} - \mathbf{x}^0) \leq f(\mathbf{x}^0) - f(\mathbf{x}^0) - \mathbf{n}^0 \cdot (\mathbf{x}^0 - \mathbf{x}^0) = 0.$$

The hyperplane $H(\mathbf{x}^0, \mathbf{n}^0)$ is valid if and only if the optimization problem (5.9) has optimal value 0. Note that by assumption, every $\mathbf{x} \in \check{X}^g$ satisfies (5.8) . The hyperplane $H(\mathbf{x}^0, \mathbf{n}^0)$ is valid if the inequality holds also for every $\mathbf{x} \in X \setminus \check{X}^g$. We assume there exists an $\mathbf{x}^- \in X \setminus \check{X}^g$ with

$$f(\mathbf{x}^-) < f(\mathbf{x}^0) + \mathbf{n}^0 \cdot (\mathbf{x}^- - \mathbf{x}^0).$$

Then the optimization problem (5.9) possesses an optimal solution $(\mathbf{x}^*, z^*)$ with value < 0, i.e.,

$$f(\mathbf{x}^*) < f(\mathbf{x}^0) + \mathbf{n}^0 \cdot (\mathbf{x}^* - \mathbf{x}^0).$$

Note that $\mathbf{x}^*$ is a globally convex domain point since $H(\mathbf{x}^*, \mathbf{n}^0)$ is valid. This is a contradiction to (5.8) for every $\mathbf{x} \in \check{X}^g$. It implies then $\mathbf{x}^-$ does not exist and (5.8) is also satisfied for every $\mathbf{x} \in X \setminus \check{X}^g$. Hence $H(\mathbf{x}^0, \mathbf{n}^0)$ is a valid hyperplane. □

Proof idea of direction "$\Leftarrow$" of Theorem 5.12. Recall Example 5.11. As shown in Figure 5.4b, for $\mathcal{S}_d$, $(\mathbf{x}_d^0, f_d(\mathbf{x}_d^0)) = (1, -536)$ is globally convex. To show Theorem 5.12, we need to prove that there exists a valid hyperplane through $(\mathbf{x}^0, f(\mathbf{x}^0)) = (1, 4, -536)$, which is the red point in Figure 5.4a. According to Corollary 5.13, every such hyperplane should contain $T_{I_{\mathbf{x}^0}}(\mathbf{x}^0)$ which is the green line in Figure 5.4a. Note that in this example $T_{I_{\mathbf{x}^0}}(\mathbf{x}^0)$ is an affine set of dimension 1. According to Lemma 5.19, by setting $P^{n-1} = T_{I_{\mathbf{x}^0}}(\mathbf{x}^0)$, for any subset $X' \subset X$ with $X' \cap F_{\mathbf{x}^0} = \emptyset$, there exists a hyperplane $H \supset T_{I_{\mathbf{x}^0}}(\mathbf{x}^0)$ such that H is below $\mathcal{S}$ over X'.

The proof idea is as follows. After finding a set P^{n-1} as described in Lemma 5.19, we split X into two compact sets X^M and X^Δ such that $\pi_{\mathbf{x}}(P^{n-1}) \subset X^M$ and $X^\Delta \cap \pi_{\mathbf{x}}(P^{n-1}) = \emptyset$ as well as $X = X^M \cup X^\Delta$. After that we first find $H^* \supset P^{n-1}$ such that H^* is below $\mathcal{S}$ over X^M. In addition, we find $H^{**} \supset P^{n-1}$ such that H^{**} is below $\mathcal{S}$ over X^Δ. According to Lemma 5.18, H^* is below H^{**} over X or H^{**} is below H^* over X. If H^* is below H^{**} over X, then H^* is below $\mathcal{S}$ over X^Δ. Together with H^* is below $\mathcal{S}$ over X^M, H^* is then below $\mathcal{S}$ over $X^M \cup X^\Delta$, hence it is valid. Otherwise if H^{**} is below H^* over X, H^{**} is then valid.

The following lemma helps us to find X^M and H^* as mentioned in the proof idea.

Lemma 5.21
For every $\mathbf{x}^0 \in \check{X}^g$, there exists an $\varepsilon > 0$ such that $T(\mathbf{x}^0)$ is below $\mathcal{S}$ over $\check{X}^g \cap B_\varepsilon(\mathbf{x}^0)$. Furthermore, every valid hyperplane H through $(\mathbf{x}^0, f(\mathbf{x}^0))$ is below $T(\mathbf{x}^0)$ over X.

Proof. Note that for any $\mathbf{x}^0 \in \operatorname{int} X$ this lemma is clear due to Lemma 5.5 and Theorem 5.8. We need only to consider the case $\mathbf{x}^0 \in \partial X$. For a given globally convex domain point $\mathbf{x}^0 \in \partial X \cap \check{X}^g$, we want to capture the globally convex domain points in its neighborhood.

Let $A, B \subset \mathbb{R}^{n+1}$ be two subsets, then

$$\pi_{\mathbf{x}}(A \cap B) = \pi_{\mathbf{x}}(A) \cap \pi_{\mathbf{x}}(B) \tag{5.10}$$

if $\pi_z(A) = \mathbb{R}$ or $\pi_z(B) = \mathbb{R}$.

For a given $\mathbf{x}^0 \in \partial X$, every subtangent plane $T_J(\mathbf{x}^0)$, $J \subset I_{\mathbf{x}^0}$ fulfills

$$\pi_{\mathbf{x}}(T_J(\mathbf{x}^0)) = \left(\bigcap_{j \in J} \{\mathbf{x} \mid a_j^T \mathbf{x} = b_j\} \right) \cap \underbrace{\pi_{\mathbf{x}}(T(\mathbf{x}^0))}_{=\mathbb{R}^n} = \bigcap_{j \in J} \{\mathbf{x} \mid a_j^T \mathbf{x} = b_j\}$$

by using (5.10) iteratively.

It implies that $\pi_{\mathbf{x}}(T_J(\mathbf{x}^0)) \cap X$ is a face of X for $J \neq \emptyset$, denoting by

$$F_J(\mathbf{x}^0) := \pi_{\mathbf{x}}(T_J(\mathbf{x}^0)) \cap X.$$

We define additionally $F_\emptyset(\mathbf{x}^0) := \pi_{\mathbf{x}}(T(\mathbf{x}^0)) \cap X = X$.

For every $I \subset I_{\mathbf{x}^0}$ we have

$$T_I(\mathbf{x}^0) = T(\mathbf{x}^0) \cap \{(\mathbf{x}, z) \mid a_i^T \mathbf{x} = b_i, i \in I\} = T(\mathbf{x}^0) \cap \{(\mathbf{x}, z) \mid \mathbf{x} \in \pi_{\mathbf{x}}(T_I(\mathbf{x}^0))\}$$

which follows directly from the definition of subtangent planes.

For any $T_I(\mathbf{x}^0)$ which is below $\mathcal{S}$, $T(\mathbf{x}^0)$ is then below $\mathcal{S}$ over $\check{X}^g \cap B_\varepsilon(\mathbf{x}^0) \cap \pi_{\mathbf{x}}(T_I(\mathbf{x}^0))$ for any $\varepsilon > 0$. It implies that $T(\mathbf{x}^0)$ is below $\mathcal{S}$ over

$$\bigcup_{\substack{I \subset I_{\mathbf{x}^0} \\ T_I(\mathbf{x}^0) \text{ is below } \mathcal{S}}} \check{X}^g \cap B_\varepsilon(\mathbf{x}^0) \cap \pi_{\mathbf{x}} \left(T_I(\mathbf{x}^0)\right).$$

Note that

$$\check{X}^g \cap B_\varepsilon(\mathbf{x}^0) = \bigcup_{I \subset I_{\mathbf{x}^0}} \check{X}^g \cap B_\varepsilon(\mathbf{x}^0) \cap \pi_{\mathbf{x}} \left(T_I(\mathbf{x}^0)\right).$$

In the following we need only to prove that for a sufficiently small $\varepsilon > 0$ it holds

$$\bigcup_{\substack{I \subset I_{\mathbf{x}^0} \\ T_I(\mathbf{x}^0) \text{ is below } \mathcal{S}}} \check{X}^g \cap B_\varepsilon(\mathbf{x}^0) \cap \pi_{\mathbf{x}} \left(T_I(\mathbf{x}^0)\right) = \bigcup_{I \subset I_{\mathbf{x}^0}} \check{X}^g \cap B_\varepsilon(\mathbf{x}^0) \cap \pi_{\mathbf{x}} \left(T_I(\mathbf{x}^0)\right) \tag{5.11}$$

which implies the first consequence that $T(\mathbf{x}^0)$ is below $\mathcal{S}$ over $\check{X}^g \cap B_\varepsilon(\mathbf{x}^0)$.

Before proving (5.11) we first prove that for every $\mathbf{x}^0 \in \check{X}^g$ there exists an $\varepsilon_J > 0$ such that any subtangent plane $T_J(\mathbf{x}^0)$, $J \subset I_{\mathbf{x}^0}$ that *does not fulfill* the condition $T_J(\mathbf{x}^0)$ is below $\mathcal{S}$ satisfies

$$\check{X}^g \cap B_{\varepsilon_J}(\mathbf{x}^0) \cap \operatorname{relint} \left(F_J(\mathbf{x}^0)\right) = \emptyset. \tag{5.12}$$

Assume that there does not exist an ε_J for which (5.12) holds. Then there exists $J \subset I_{\mathbf{x}^0}$ such that $T_J(\mathbf{x}^0)$ does not fulfill $T_J(\mathbf{x}^0)$ is below $\mathcal{S}$ and for every $\varepsilon_J > 0$

$$\check{X}^g \cap B_{\varepsilon_J}(\mathbf{x}^0) \cap \operatorname{relint} \left(F_J(\mathbf{x}^0)\right) \neq \emptyset.$$

Recall that $F_J(\mathbf{x}^0)$ is a compact set. There exists then a point sequence $(\mathbf{x}^i)_{i\in\mathbb{N}} \subset \check{X}^g$, as shown in Figure 5.8, such that

$$\lim_{i\to\infty} \mathbf{x}^i = \mathbf{x}^0 \text{ and } \{((\mathbf{x}^i)_{i\in\mathbb{N}})\} \subset \left(\check{X}^g \cap B_{\varepsilon_J}(\mathbf{x}^0) \cap \operatorname{relint}\left(F_J(\mathbf{x}^0)\right)\right)$$

for $\varepsilon_J > 0$. The point $\mathbf{x}^i$ for any $i \in \mathbb{N}$ is a globally convex domain point and a relative interior point in $F_J(\mathbf{x}^0)$. Denoting $\mathbf{x}^i$ by $\mathbf{y}$, the function

$$G_{\mathbf{y}}(\mathbf{x}) = f(\mathbf{x}) - (\nabla f(\mathbf{y}) \cdot (\mathbf{x} - \mathbf{y}) + f(\mathbf{y}))$$

evaluates the validity of the tangent plane $T(\mathbf{y})$ through $(\mathbf{y}, f(\mathbf{y}))$: Tangent plane $T(\mathbf{y})$ is valid if and only if $G_{\mathbf{y}}(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in X$; subtangent plane $T_J(\mathbf{y})$ is below $\mathcal{S}$ if and only if $G_{\mathbf{y}}(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in F_J(\mathbf{y})$. Consider the function

$$h : F_J(\mathbf{x}^0) \to \mathbb{R}, \mathbf{y} \mapsto \min_{\mathbf{x}\in F_J(\mathbf{y})} G_{\mathbf{y}}(\mathbf{x}).$$

With "⇒" of Theorem 5.12, $\mathbf{y} \in \check{X}^g$ implies that $T_J(\mathbf{y})$ is below $\mathcal{S}$ which is equivalent to $T(\mathbf{y})$ is below $\mathcal{S}$ over $F_J(\mathbf{y})$. Thus, $h(\mathbf{x}^i) \geq 0$ for every $i \in \mathbb{N}$. For any $\mathbf{y}$ function $G_{\mathbf{y}}(\mathbf{x})$ is a continuous function, it implies h is also continuous since it maps from a compact set to the minimum of $G_{\mathbf{y}}(\mathbf{x})$. Together with $\lim_{i \to \infty} \mathbf{x}^i = \mathbf{x}^0$ it implies that $h(\mathbf{x}^0) \geq 0$. This implies that $T_J(\mathbf{x}^0)$ is below $\mathcal{S}$. This is a contradiction to our assumption. Hence there exists an $\varepsilon_J > 0$ such that

$$\check{X}^g \cap B_{\varepsilon_J}(\mathbf{x}^0) \cap \operatorname{relint}\left(F_J(\mathbf{x}^0)\right) = \emptyset.$$

Since we have finitely many subtangent planes through $(\mathbf{x}^0, f(\mathbf{x}^0))$, there exists an $\varepsilon > 0$ which is the minimum of all ε_J such that (5.12) holds for every such subtangent plane through $(\mathbf{x}^0, f(\mathbf{x}^0))$.

Recall (5.11) that we are now ready to prove. For any $J \subset I_{\mathbf{x}^0}$ with $J \neq I_{\mathbf{x}^0}$, we study now the point set $F_J(\mathbf{x}^0) \cap B_\varepsilon(\mathbf{x}^0)$. For any $\mathbf{x} \in F_J(\mathbf{x}^0) \cap B_\varepsilon(\mathbf{x}^0)$ we have two cases, either

$$\mathbf{x} \in \operatorname{relint}\left(F_J(\mathbf{x}^0) \cap B_\varepsilon(\mathbf{x}^0)\right)$$

or

$$\mathbf{x} \in \partial\left(F_J(\mathbf{x}^0) \cap B_\varepsilon(\mathbf{x}^0)\right).$$

Since $J \subset I_{\mathbf{x}^0}$ with $J \neq I_{\mathbf{x}^0}$, we have $I_{\mathbf{x}^0} \setminus J \neq \emptyset$. For any $\mathbf{x} \in \partial\left(F_J(\mathbf{x}^0) \cap B_\varepsilon(\mathbf{x}^0)\right)$, there exists $i \in I_{\mathbf{x}^0} \setminus J$ such that $\mathbf{x} \in \left(\pi_{\mathbf{x}}(T_{J'}(\mathbf{x}^0)) \cap X \cap B_\varepsilon(\mathbf{x}^0)\right)$ with $J' = J \cup \{i\}$. Together with $\pi_{\mathbf{x}}(T_{J'}(\mathbf{x}^0)) \subset \pi_{\mathbf{x}}(T_J(\mathbf{x}^0))$ for any $J \subset J'$ and the definition

$$S_J^\varepsilon(\mathbf{x}^0) := F_J(\mathbf{x}^0) \cap B_\varepsilon(\mathbf{x}^0),$$

it implies then

$$S_J^\varepsilon(\mathbf{x}^0) = \operatorname{relint}\left(S_J^\varepsilon(\mathbf{x}^0)\right) \cup \underbrace{\left(\bigcup_{\substack{i \in I_{\mathbf{x}^0} \setminus J \\ J' = J \cup \{i\}}} \left(S_{J'}^\varepsilon(\mathbf{x}^0)\right)\right)}_{= \partial S_J^\varepsilon(\mathbf{x}^0)}. \tag{5.13}$$

Note that for any sets A, B, C it holds

$$C \cap (A \cup B) = (C \cap A) \cup (C \cap B).$$

For any T_J which does not fulfill T_J is below $\mathcal{S}$, we have then

$$S_J^\varepsilon(\mathbf{x}^0) \cap \check{X}^g = \underbrace{\left(\operatorname{relint}\left(S_J^\varepsilon(\mathbf{x}^0)\right) \cap \check{X}^g\right)}_{\overset{(5.12)}{=} \emptyset} \cup \left(\bigcup_{\substack{i \in I_{\mathbf{x}^0} \setminus J \\ J' = J \cup \{i\}}} \left(S_{J'}^\varepsilon(\mathbf{x}^0) \cap \check{X}^g\right)\right) \tag{5.14}$$

Using (5.14) iteratively and together with (5.12), it implies that

$$\check{X}^g \cap S_J^\varepsilon(\mathbf{x}^0) = \bigcup_{\substack{J \subset I \subset I_{\mathbf{x}^0} \\ T_I(\mathbf{x}^0) \text{ is below } \mathcal{S}}} \check{X}^g \cap S_I^\varepsilon(\mathbf{x}^0)$$

for every $J \subset I_{\mathbf{x}^0}$. This implies $T(\mathbf{x}^0)$ is below $\mathcal{S}$ over

$$\bigcup_{\substack{I \subset I_{\mathbf{x}^0} \\ T_I(\mathbf{x}^0) \text{ is below } \mathcal{S}}} \check{X}^g \cap S_I^\varepsilon(\mathbf{x}^0) = \bigcup_{I \subset I_{\mathbf{x}^0}} \check{X}^g \cap S_I^\varepsilon(\mathbf{x}^0) = \check{X}^g \cap B_\varepsilon(\mathbf{x}^0).$$

Now it remains only to prove that every valid hyperplane H through $(\mathbf{x}^0, f(\mathbf{x}^0))$ is below $T(\mathbf{x}^0)$ over X. Assume that $H \neq T(\mathbf{x}^0)$, otherwise we are done. An example for $n = 1$ is shown in Figure 5.9. Let $X = [l, u]$ and l, x^m and u be globally convex points. Tangent plane $T(l)$, shown as the green line, is not valid. However, the example in Figure 5.9 shows that every valid hyperplane through $(l, f(l))$ is below $T(l)$ over X; it also shows that every hyperplane through $(l, f(l))$ which is above $T(l)$ over X cannot be valid. For interior domain point x^m, there does not exist another hyperplane through $(x^m, f(x^m))$ which is below $T(x^m)$, thus there exists at most one valid hyperplane through $(x^m, f(x^m))$ which is also implied by Theorem 5.8. For domain point u, $T(u)$ is valid. In addition, every hyperplane through $(u, f(u))$ which is below $T(u)$ over X is also valid; every hyperplane through $(u, f(u))$ which is not $T(u)$ and above $T(u)$ over X is not valid.

We go back to the proof for general n. Under the assumption $H \neq T(\mathbf{x}^0)$, hyperplanes H and $T(\mathbf{x}^0)$ are not parallel due to the intersection point $(\mathbf{x}^0, f(\mathbf{x}^0))$. Similar to the notation in Lemma 5.18, denote

$$\pi_{\mathbf{x}}(H \cap T(\mathbf{x}^0)) = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{n}^0 \cdot (\mathbf{x} - \mathbf{x}^0) = 0\},$$

and

$$L := \{\mathbf{x} \in \mathbb{R}^n | \mathbf{n}^0 \cdot (\mathbf{x} - \mathbf{x}^0) \leq 0\}, R := \{\mathbf{x} \in \mathbb{R}^n | \mathbf{n}^0 \cdot (\mathbf{x} - \mathbf{x}^0) \geq 0\}$$

with $\mathbf{n}^0 \in \mathbb{R}^n$. Note that since $\mathbf{x}^0$ is a boundary point of X, we have either $\pi_{\mathbf{x}}(H \cap T(\mathbf{x}^0)) \cap \operatorname{int} X \neq \emptyset$ or $\pi_{\mathbf{x}}(H \cap T(\mathbf{x}^0)) \cap \operatorname{int} X = \emptyset$. If $\pi_{\mathbf{x}}(H \cap T(\mathbf{x}^0)) \cap \operatorname{int} X \neq \emptyset$, then we have $L \cap \operatorname{int} X \neq \emptyset$ and $R \cap \operatorname{int} X \neq \emptyset$. Lemma 5.18 implies H is above $T(\mathbf{x}^0)$ over L or R. Without loss of generality, H is above $T(\mathbf{x}^0)$ over L. Consider the graph of f over $L \cap B_\varepsilon(\mathbf{x}^0) \cap \operatorname{int} X \neq \emptyset$ for any $\varepsilon > 0$. There always exists a point $\mathbf{x}^1 \in L \cap B_\varepsilon(\mathbf{x}^0) \cap \operatorname{int} X$ such that $(\mathbf{x}^1, f(\mathbf{x}^1))$ is below H. Thus H cannot be valid if $\pi_{\mathbf{x}}(H \cap T(\mathbf{x}^0)) \cap \operatorname{int} X \neq \emptyset$.

Now we consider the case $\pi_{\mathbf{x}}(H \cap T(\mathbf{x}^0)) \cap \operatorname{int} X = \emptyset$. There are two cases, either $X \subset L$ or $X \subset R$. Without loss of generality, let $X \subset L$. Lemma 5.18 implies that H is either below $T(\mathbf{x}^0)$ or above $T(\mathbf{x}^0)$ over X. If H is above $T(\mathbf{x}^0)$ over X, as we discussed above, using Taylor's formula, there always exists a point $\mathbf{x}^1 \in B_\varepsilon(\mathbf{x}^0) \cap \operatorname{int} X$ such that $(\mathbf{x}^1, f(\mathbf{x}^1))$ is below H. Thus H cannot be valid if H is above $T(\mathbf{x}^0)$ over X. □

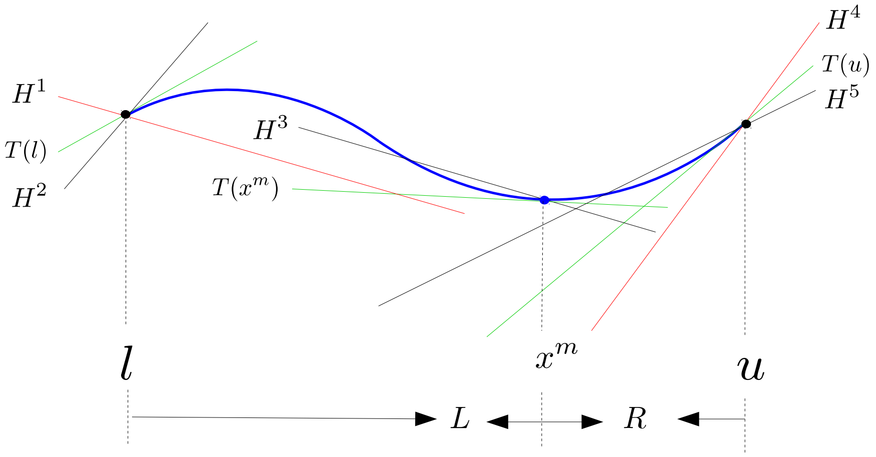


Figure 5.9: Examples to compare hyperplanes and the tangent plane both through a common point

Now we are ready to prove the direction "⇐" of Theorem 5.12.

Proof (of "⇐" of Theorem 5.12). For $\mathbf{x}^0 \in \partial X$, as we discussed before, there exists a smallest face $F_{\mathbf{x}^0} \subset X$ with $\mathbf{x}^0 \in F_{\mathbf{x}^0}$. At the end of the proof of "⇒" we showed that $\mathbf{x}^0_d = g_d(\mathbf{x}^0)$ is a globally convex domain point of $\mathcal{S}_d$ if and only if $T_{I_{\mathbf{x}^0}}(\mathbf{x}^0)$ is below $\mathcal{S}$ over $F_{\mathbf{x}^0}$. We want to show that there exists a valid hyperplane $H \supset T_{I_{\mathbf{x}^0}}(\mathbf{x}^0)$. An example can be seen in Figure 5.10.

Since $F_{\mathbf{x}^0}$ is a face of X, there exists vector $C \in \mathbb{R}^n \setminus \{0\}$ such that

$$F_{\mathbf{x}^0} = \operatorname{argmin}\left\{C^T\mathbf{x} \mid \mathbf{x} \in X\right\},$$

see [Sch86]. We define the affine set

$$X^{n-1}_{\mathbf{x}^0} = \{\mathbf{x} \mid C^T\mathbf{x} - C^T\mathbf{x}^0 = 0, \mathbf{x} \in \mathbb{R}^n\} \subset \mathbb{R}^n$$

with $F_{\mathbf{x}^0} \subset X^{n-1}_{\mathbf{x}^0}$ and dimension $n-1$ and the set

$$X^L = \{\mathbf{x} \mid C^T\mathbf{x} - C^T\mathbf{x}^0 \geq 0, \mathbf{x} \in \mathbb{R}^n\}$$

which satisfies $X^L \supset X$.

Besides $(\mathbf{x}^0, f(\mathbf{x}^0))$, the subtangent plane $T_{I_{\mathbf{x}^0}}(\mathbf{x}^0)$ may have other intersection points with $\mathcal{S}$, we denote the set of all intersection points by

$$\mathcal{S}_{\mathcal{T}_{\mathbf{x}^0}} := \mathcal{S} \cap T_{I_{\mathbf{x}^0}}(\mathbf{x}^0) = \{(\mathbf{x}, z) \mid z = f(\mathbf{x}), \mathbf{x} \in F_{\mathbf{x}^0}\} \cap T(\mathbf{x}^0)$$

with the corresponding domain set $X_{\mathcal{T}_{\mathbf{x}^0}} = \pi_{\mathbf{x}}(\mathcal{S}_{\mathcal{T}_{\mathbf{x}^0}})$. For any $\mathbf{x}^1 \in X_{\mathcal{T}_{\mathbf{x}^0}}, \mathbf{x}^1 \neq \mathbf{x}^0$ with $(\mathbf{x}^1, f(\mathbf{x}^1)) \in T(\mathbf{x}^0)$, we want to prove $T(\mathbf{x}^0)$ is below $T(\mathbf{x}^1)$ over $F_{\mathbf{x}^0}$. Let $g_0 : \mathbb{R}^n \to \mathbb{R}$ and $g_1 : \mathbb{R}^n \to \mathbb{R}$ be two affine functions such that

$$T(\mathbf{x}^0) = \{(\mathbf{x}, g_0(\mathbf{x})) \mid \mathbf{x} \in \mathbb{R}^n\} \text{ and } T(\mathbf{x}^1) = \{(\mathbf{x}, g_1(\mathbf{x})) \mid \mathbf{x} \in \mathbb{R}^n\}.$$

Assume that the statement $T(\mathbf{x}^0)$ is below $T(\mathbf{x}^1)$ over $F_{\mathbf{x}^0}$ is not true, then for any $\varepsilon_1 > 0$ there exists

$$\mathbf{x}^2 \in B_{\varepsilon_1}(\mathbf{x}^1) \cap F_{\mathbf{x}^0}$$

with $g_0(\mathbf{x}^2) < g_1(\mathbf{x}^2)$. Using "Taylor's Formula in Several Variables"(see the book [Edw94]), there exists $\mathbf{x}^3$ which is a linear combination of $\mathbf{x}^1$ and $\mathbf{x}^2$ such that $g_0(\mathbf{x}^3) > f(\mathbf{x}^3)$. This is a contradiction to the condition that $T(\mathbf{x}^0)$ is below $\mathcal{S}$ over $F_{\mathbf{x}^0}$. Hence $T(\mathbf{x}^0)$ is below $T(\mathbf{x}^1)$ over $F_{\mathbf{x}^0}$.

Consider the affine set

$$P_H^{n-1} = T(\mathbf{x}^0) \cap \{(\mathbf{x}, z) \mid \mathbf{x} \in X_{\mathbf{x}^0}^{n-1}, z \in \mathbb{R}\}.$$

For any point $(\mathbf{x}^f, z^f)$ with $\mathbf{x}^f \in X \setminus F_{\mathbf{x}^0}$, aff$\{P_H^{n-1}, \{(\mathbf{x}^f, z^f)\}\}$ is a nonvertical hyperplane. Consider the linear convex set $\{(\mathbf{x}, z) \in T(\mathbf{x}^1) \mid \mathbf{x} \in X, z \in \mathbb{R}\}$. For any $\mathbf{x}^1 \in X_{\mathcal{T}_{\mathbf{x}^0}} \setminus \{\mathbf{x}^0\}$, there exists a z_1^f such that aff$\{P_H^{n-1}, \{(\mathbf{x}^f, z_1^f)\}\}$ is below $T(\mathbf{x}^1)$ over X. As a consequence, there exists z^f such that $H^* = \text{aff}\{P_H^{n-1}, \{(\mathbf{x}^f, z^f)\}\}$ is below $T(\mathbf{x}^*)$ over X for any $\mathbf{x}^* \in X_{\mathcal{T}_{\mathbf{x}^0}}$. Due to Lemma 5.21, there exists a sufficiently small ε^1 such that H^* is below $\mathcal{S}$ over $\check{X}^g \cap B_{\varepsilon^1}(\mathbf{x}^*)$ for any $\mathbf{x}^* \in X_{\mathcal{T}_{\mathbf{x}^0}}$. As a result, H^* is below $\mathcal{S}$ over

$$\bigcup_{\mathbf{x}^* \in X_{\mathcal{T}_{\mathbf{x}^0}}} \check{X}^g \cap B_{\varepsilon^1}(\mathbf{x}^*).$$

For the set

$$X^{M_1} = \bigcup_{\mathbf{x}^* \in X_{\mathcal{T}_{\mathbf{x}^0}}} B_{\varepsilon^1}(\mathbf{x}^*),$$

the point $(\mathbf{x}^2, f(\mathbf{x}^2))$ is below $T_{I_{\mathbf{x}^0}}(\mathbf{x}^0)$ for any $\mathbf{x}^2 \in F_{\mathbf{x}^0} \setminus X^{M_1}$. Then there exists an ε^2 such that H^* is below $\mathcal{S}$ over $B_{\varepsilon^2}(\mathbf{x}^2)$ for any $\mathbf{x}^2 \in F_{\mathbf{x}^0} \setminus X^{M_1}$. By defining a new set

$$X^{M_2} = \left(\bigcup_{\mathbf{x}^* \in X_{\mathcal{T}_{\mathbf{x}^0}}} B_{\varepsilon^1}(\mathbf{x}^*) \right) \cup \left(\bigcup_{\mathbf{x}^2 \in F_{\mathbf{x}^0} \setminus X^{M_1}} B_{\varepsilon^2}(\mathbf{x}^2) \right),$$

as well as

$$X^M = X^{M_2} \cap X,$$

it implies that H^* is below $\mathcal{S}$ over $X^M \cap \check{X}^g$.

Consider now $X^{\Delta} = X \setminus \operatorname{int} X^{M_2}$ Then we have $X = X^{\Delta} \cup X^{M}$. Using X^{Δ} as S in Lemma 5.19, there exists a hyperplane H^{**} which is below $\mathcal{S}$ over X^{Δ}.

Finally, since we have $H^{*} \supset P_H^{n-1}$ and $H^{**} \supset P_H^{n-1}$, together with Lemma 5.18, we have two cases by comparing H^{*} and H^{**}. If H^{*} is below H^{**} over X, then H^{*} is below $\mathcal{S}$ over

$$\Big((X^{M} \cap \check{X}^{g}) \cup X^{\Delta}\Big) \supset \check{X}^{g}.$$

Due to Lemma 5.20 we have then H^{*} is a valid hyperplane through $(\mathbf{x}^0, f(\mathbf{x}^0))$. Otherwise H^{**} is below H^{*} over X. With the same reason H^{**} is a valid hyperplane through $(\mathbf{x}^0, f(\mathbf{x}^0))$. Until now we finish the proof. □

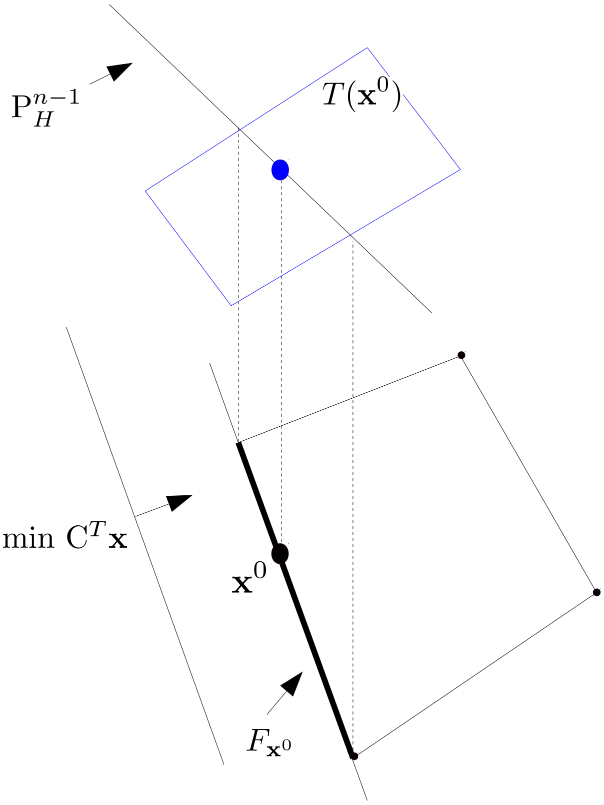


Figure 5.10: Example for the proof of Theorem 5.12

Note that for every extreme point $\mathbf{x}^e \in X^e \subset \partial X$, the corresponding $\mathbf{x}_d^e$ satisfies $\mathcal{S}_d = \{\mathbf{x}_d^e\}$. As the single point in that set, $\mathbf{x}_d^e$ is globally convex which implies the following corollary.

Corollary 5.22
Every extreme domain point $\mathbf{x}^e \in X^e$ *is globally convex.*

5.3.4 Tight and loose hyperplanes

Until now, we observed valid subtangent planes through a fixed given point $(\mathbf{x}^0, f(\mathbf{x}^0))$. From now on we restrict our attention to valid subtangent planes through $(\mathbf{x}^0, f(\mathbf{x}^0))$ that are contained in a fixed given valid hyperplane H.

Definition 5.23 (H-contained subtangent planes through $(\mathbf{x}^0, f(\mathbf{x}^0))$)
Let $\mathbf{x}^0 \in \partial X$ and H be a valid hyperplane containing $(\mathbf{x}^0, f(\mathbf{x}^0))$. A valid subtangent plane $T_I(\mathbf{x}^0) \subset H$ is said to be an *H-contained subtangent plane through* $(\mathbf{x}^0, f(\mathbf{x}^0))$. Such a $T_I(\mathbf{x}^0)$ is said to be *maximally valid* if there does not exist any other H-contained subtangent plane $T_{I'}(\mathbf{x}^0)$ through $(\mathbf{x}^0, f(\mathbf{x}^0))$ such that $T_I(\mathbf{x}^0) \subsetneq T_{I'}(\mathbf{x}^0)$.

In Example 5.15, $T_{\{1\}}(\mathbf{x}^0)$ is an H^1-contained subtangent plane, $T_{\{2\}}(\mathbf{x}^0)$ is an H^2-contained subtangent plane and $T_{\{1,2\}}(\mathbf{x}^0)$ is an H^1-contained, H^2-contained and H^3-contained subtangent plane.

We may wonder for which faces $F \ni \mathbf{x}^0$ of X there exists a valid hyperplane H such that $H \supset (T(\mathbf{x}^0) \cap \{(\mathbf{x}, z) \mid \mathbf{x} \in F\})$. We start to answer the question with the following lemma.

Lemma 5.24 (Unique maximally valid H-contained subtangent plane)
For any $\mathbf{x}^0 \in \partial X$ *with a valid hyperplane* H *through* $(\mathbf{x}^0, f(\mathbf{x}^0))$*, there always exists a unique maximally valid* H*-contained subtangent plane through* $(\mathbf{x}^0, f(\mathbf{x}^0))$*, denoted by* $T_H^{\max}(\mathbf{x}^0)$*.*

Proof. We begin the proof with a few definitions. We define first a set of index sets

$$\mathcal{I}_{\mathbf{x}^0}^H = \left\{ I \subset I_{\mathbf{x}^0} \mid T_I(\mathbf{x}^0) \subset H \right\}.$$

The set $\mathcal{I}_{\mathbf{x}^0}^H$ is not empty since we have $I_{\mathbf{x}^0} \in \mathcal{I}_{\mathbf{x}^0}^H$. In addition, we define

$$I_H^{\max} = \bigcap_{I \in \mathcal{I}_{\mathbf{x}^0}^H} I.$$

Now we want to prove that $T_{I_H^{\max}}(\mathbf{x}^0)$ is the unique maximally valid H-contained subtangent plane. On the one hand, as we proved before, for any $I_1 \subset I_{\mathbf{x}^0}, I_2 \subset I_{\mathbf{x}^0}$, it holds

$$T_{I_1 \cap I_2}(\mathbf{x}^0) = \operatorname{aff}\{T_{I_1}(\mathbf{x}^0), T_{I_2}(\mathbf{x}^0)\}.$$

Since $T_{I_1}(\mathbf{x}^0) \subset H$ and $T_{I_2}(\mathbf{x}^0) \subset H$, then we have also $T_{I_1 \cap I_2}(\mathbf{x}^0) \subset H$. Using this iteratively, it implies that $T_{I_H^{\max}}(\mathbf{x}^0) \subset H$.

One the other hand, for any $I_1 \subset I_2 \subset I_{\mathbf{x}^0}$ we have $T_{I_1}(\mathbf{x}^0) \supset T_{I_2}(\mathbf{x}^0)$. It implies then for any $I \subset I_{\mathbf{x}^0}$ with $T_I(\mathbf{x}^0) \subset H$, we have also $T_I(\mathbf{x}^0) \subset T_{I_H^{\max}}(\mathbf{x}^0)$ since $I_H^{\max} \subset I$.

As a result $T_{I_H^{\max}}(\mathbf{x}^0)$ is a maximally valid H-contained subtangent plane.

Assume there exists another maximally valid H-contained subtangent plane $T_J(\mathbf{x}^0)$. Then there exists $j \in I_H^{\max} \setminus J$. Recall the definition of $\mathcal{I}_{\mathbf{x}^0}^H$, for every $I \subset I_{\mathbf{x}^0}$ with $T_I(\mathbf{x}^0) \subset H$ we must have $j \in I$. It is a contradiction that $T_J(\mathbf{x}^0) \subset H$. Hence $T_{I_H^{\max}}(\mathbf{x}^0) =: T_H^{\max}(\mathbf{x}^0)$ is the unique maximally valid H-contained subtangent plane.

□

In Example 5.15, $T_{\{1\}}(\mathbf{x}^0)$ is the maximally valid H^1-contained subtangent plane, $T_{\{2\}}(\mathbf{x}^0)$ is the maximally valid H^2-contained subtangent plane and $T_{\{1,2\}}(\mathbf{x}^0)$ is the maximally valid H^3-contained subtangent plane. $T_{\{1,2\}}(\mathbf{x}^0)$ is an H^1-contained and H^2-contained subtangent plane, but neither is maximally valid.

In Example 5.15, H^1, H^2 and H^3 are three different valid hyperplanes with corresponding halfspaces $\check{H}^1$, $\check{H}^2$ and $\check{H}^3$. Recall that the convex hull of $\mathcal{S}$ is the intersection of all halfspaces containing $\mathcal{S}$. Observe that

$$\check{H}^3 \supset (\check{H}^1 \cap \check{H}^2) \Rightarrow (\check{H}^1 \cap \check{H}^2) = (\check{H}^1 \cap \check{H}^2 \cap \check{H}^3).$$

It implies that to form the convex hull of $\mathcal{S}$, we do not have to use $\check{H}^3$. Hence $\check{H}^3$ can be regarded as an "inefficient" halfspace. To discuss which halfspaces are efficient and which are not, we make the following definition.

Definition 5.25 (Tight and loose hyperplanes)
Let H be a valid hyperplane and $X^H := \pi_{\mathbf{x}}(H \cap \mathcal{S})$ be its associated set of domain points with $X^H \neq \emptyset$. Let

$$S^H := \operatorname{aff}\left\{T_H^{\max}(\mathbf{x}^0) : \text{ for all } \mathbf{x}^0 \in X^H\right\} \subset H$$

be the affine hull of the corresponding maximally valid subtangent planes for every $\mathbf{x}^0 \in X^H$. Note that $S^H \subset H$ with $\dim S^H \leq n$ since $T_H^{\max}(\mathbf{x}^0) \subset H$ for any $\mathbf{x}^0 \in X^H$. The hyperplane H is said to be a *tight* hyperplane if $\dim S^H = n$. Otherwise, in case of $\dim S^H < n$, H is said to be a *loose* hyperplane. The corresponding valid halfspace is also said to be tight or loose, respectively.

For every tight hyperplane H we have $S^H = H$; for every loose hyperplane H we have $S^H \subsetneq H$. Then we can also compute the dimension of S^H by using the following lemma.

Lemma 5.26
Let H be a valid hyperplane, define

$$X_H^e = \bigcup_{\mathbf{x}^0 \in \pi_{\mathbf{x}}(H \cap \mathcal{S})} \underbrace{\pi_{\mathbf{x}}\left(T(\mathbf{x}^0) \cap H\right) \cap X^e}_{=:X^e_{H(\mathbf{x}^0)}},$$

where X^e is the set of all extreme points of X. Then we have

$$\dim S^H = \dim \operatorname{aff}\{X_H^e\}.$$

Consider Example 5.15 again before we start the proof. Define four points $\mathbf{x}^0 = (-3,-3)$, $\mathbf{x}^1 = (10,10)$, $\mathbf{x}^2 = (-3,10)$ and $\mathbf{x}^3 = (10,-3)$. Then we have $X^e = \{\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3\}$. Note that $\pi_{\mathbf{x}}(H^i \cap \mathcal{S}) = \{\mathbf{x}^0, \mathbf{x}^1\}$ for $i = 1, 2, 3$. However, the computations

$$\begin{aligned}
X^e_{H^1} &= X^e_{H^1(\mathbf{x}^0)} \cup X^e_{H^1(\mathbf{x}^1)} = \{\mathbf{x}^0, \mathbf{x}^2\} \cup \{\mathbf{x}^1\} = \{\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2\}, \\
X^e_{H^2} &= X^e_{H^2(\mathbf{x}^0)} \cup X^e_{H^2(\mathbf{x}^1)} = \{\mathbf{x}^0, \mathbf{x}^3\} \cup \{\mathbf{x}^1\} = \{\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^3\}, \\
X^e_{H^3} &= X^e_{H^3(\mathbf{x}^0)} \cup X^e_{H^3(\mathbf{x}^1)} = \{\mathbf{x}^0\} \cup \{\mathbf{x}^1\} = \{\mathbf{x}^0, \mathbf{x}^1\}.
\end{aligned}$$

show that H^1 and H^2 are tight but H^3 is loose.

Proof. It is clear that $S^H \subset H$ and H are nonvertical. It implies that

$$\dim S^H = \dim \pi_{\mathbf{x}}(S^H) = \operatorname{aff}\left\{\pi_{\mathbf{x}}(T^{\max}_H(\mathbf{x}^0)) : \text{ for all } \mathbf{x}^0 \in X^H\right\}.$$

We prove

$$\pi_{\mathbf{x}}(T^{\max}_H(\mathbf{x}^0)) \cap X = \pi_{\mathbf{x}}(T(\mathbf{x}^0) \cap H) \cap X.$$

From the definition of $T^{\max}_H(\mathbf{x}^0)$ we know that there exists $I^{\max} \subset I_{\mathbf{x}^0}$ with

$$T^{\max}_H(\mathbf{x}^0) = T_{I^{\max}}(\mathbf{x}^0) = T(\mathbf{x}^0) \cap \left(\bigcap_{j \in I^{\max}} \left\{(\mathbf{x}, z) \mid a_j^T \mathbf{x} = b_j\right\}\right).$$

Using (5.10) recursively, we have

$$\pi_{\mathbf{x}}(T^{\max}_H(\mathbf{x}^0)) = \left(\bigcap_{j \in I^{\max}} \{\mathbf{x} \mid a_j^T \mathbf{x} = b_j\}\right) \cap \underbrace{\pi_{\mathbf{x}}(T(\mathbf{x}^0))}_{=\mathbb{R}^n} = \bigcap_{j \in I^{\max}} \{\mathbf{x} \mid a_j^T \mathbf{x} = b_j\},$$

which implies that $\pi_{\mathbf{x}}(T^{\max}_H(\mathbf{x}^0)) \cap X$ is a face of X. Note that we have

$$\pi_{\mathbf{x}}(T^{\max}_H(\mathbf{x}^0)) \cap X \subset \pi_{\mathbf{x}}(T(\mathbf{x}^0) \cap H) \cap X,$$

since $T^{\max}_H(\mathbf{x}^0) \subset (T(\mathbf{x}^0) \cap H)$. Assume that there exists $\mathbf{x}^1 \in \pi_{\mathbf{x}}(T(\mathbf{x}^0) \cap H) \cap X$ but $\mathbf{x}^1 \notin \pi_{\mathbf{x}}(T^{\max}_H(\mathbf{x}^0))$. It implies that there exists another face

$$F^1 = \operatorname{aff}\left\{\bigcap_{j \in I^{\max}} \left\{(\mathbf{x}, z) \mid a_j^T \mathbf{x} = b_j\right\}, \{\mathbf{x}^1\}\right\} \cap X$$

of X with $\pi_{\mathbf{x}}(T^{\max}_H(\mathbf{x}^0)) \subsetneq F^1$ and $F^1 \subset \pi_{\mathbf{x}}(T(\mathbf{x}^0) \cap H)$. Then there exists a subtangent plane $T_I(\mathbf{x}^0)$ with $\pi_{\mathbf{x}}(T_I(\mathbf{x}^0)) = F^1$ and $T^{\max}_H(\mathbf{x}^0) \subsetneq T_I(\mathbf{x}^0) \subset H$. This is a contradiction to $T^{\max}_H(\mathbf{x}^0)$ is the maximally valid H-contained subtangent plane. Hence we have

$$\pi_{\mathbf{x}}(T^{\max}_H(\mathbf{x}^0)) \cap X = \pi_{\mathbf{x}}(T(\mathbf{x}^0) \cap H) \cap X.$$

As we discussed, $\pi_{\mathbf{x}}(T(\mathbf{x}^0) \cap H) \cap X$ is a face of X, which implies

$$\pi_{\mathbf{x}}(T(\mathbf{x}^0) \cap H) \cap X = \operatorname{conv}\{\pi_{\mathbf{x}}(T(\mathbf{x}^0) \cap H) \cap X \cap X^e\} = \operatorname{conv}\{\pi_{\mathbf{x}}(T(\mathbf{x}^0) \cap H) \cap X^e\}.$$

We then have

$$\operatorname{aff}\{\pi_{\mathbf{x}}(T_H^{\max}(\mathbf{x}^0))\} = \operatorname{aff}\{\pi_{\mathbf{x}}(T_H^{\max}(\mathbf{x}^0)) \cap X\} = \operatorname{aff}\{\pi_{\mathbf{x}}(T(\mathbf{x}^0) \cap H) \cap X^e\}$$

for every $\mathbf{x}^0 \in X^H = \pi_{\mathbf{x}}(H \cap \mathcal{S})$. It implies

$$\begin{aligned}\dim S^H &= \dim \operatorname{aff}\left\{\pi_{\mathbf{x}}(T_H^{\max}(\mathbf{x}^0)) : \text{ for all } \mathbf{x}^0 \in \pi_{\mathbf{x}}(H \cap \mathcal{S})\right\} \\ &= \dim \operatorname{aff}\left\{\bigcup_{\mathbf{x}^0 \in \pi_{\mathbf{x}}(H \cap \mathcal{S})} \pi_{\mathbf{x}}\left(T(\mathbf{x}^0) \cap H\right) \cap X^e\right\} = \dim \operatorname{aff}\{X_H^e\}.\end{aligned}$$

□

5.3.5 Extendability

Recall Example 5.15 again. Halfspaces $\check{H}^1$ and $\check{H}^2$ are tight and $\check{H}^3$ is loose. We can check that $\check{H}^3 \subsetneq (\check{H}^1 \cup \check{H}^2)$ which implies that $\check{H}^3$ is redundant to form the convex hull of $\mathcal{S}$. We may have the intuition that to form the convex hull of $\mathcal{S}$ by intersecting halfspaces containing $\mathcal{S}$, we need only to choose the tight ones. We will verify this hypothesis later.

The following theorem provides a relation between loose and tight hyperplanes.

Theorem 5.27 (Extendability)
Let H^l be a loose hyperplane. Then there exists a tight hyperplane H^t such that

$$X^{H^l} \subset X^{H^t},$$

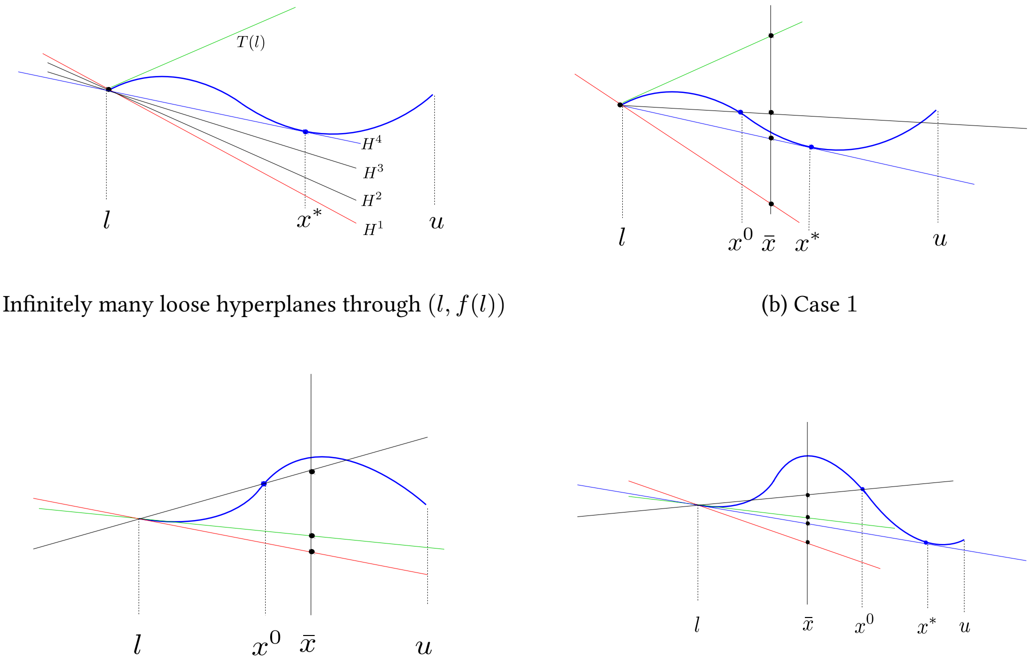
where $X^{H^l} := \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$ and $X^{H^t} := \pi_{\mathbf{x}}(H^t \cap \mathcal{S})$.

Before we prove the theorem, consider first the simplest case of dimension $n = 1$. Let $X = [l, u]$. Note that X has only two boundary domain points. A valid hyperplane can only be loose if it intersects the graph of f either at $(l, f(l))$ or at $(u, f(u))$. An example is shown in Figure 5.11a, H^1, H^2 and H^3 are loose hyperplanes through $(l, f(l))$ while H^4 is their corresponding tight hyperplane fulfilling Theorem 5.27 since

$$1 \geq \dim S^{H^4} \geq \dim \operatorname{aff}\{l, x^*\} = 1$$

implies $\dim S^{H^4} = 1$. Actually there are infinitely many loose hyperplanes through $(l, f(l))$.

Before we give the details for the proof of Theorem 5.27 for the general case in space $\mathbb{R}^{n+1}$, we prove first the special case of $n = 1$. With this proof we illustrate the proof idea for the general case. We give the proof at the end after a few preliminaries.



(a) Infinitely many loose hyperplanes through $(l, f(l))$

(b) Case 1

(c) Case $2a$

(d) Case $2b$

Figure 5.11: Example for the proof of Theorem 5.27 for $n = 1$: loose hyperplanes through $(l, f(l))$ in dimension $n = 1$ and the corresponding tight hyperplanes

Proof (of Theorem 5.27 for $n = 1$). For any interior domain point $x \in (l, u)$, if a hyperplane $H \ni (x, f(x))$ is valid then Lemma 5.5 implies that $H = T(x)$. It is clear that $T(x)$ is tight if it is valid, thus we only need to consider the boundary domain points. Without loss of generality we consider only loose hyperplanes through $(l, f(l))$. Actually we need only to prove that there always exists a tight hyperplane through $(l, f(l))$. Consider the three different examples shown in Figure 5.11b, Figure 5.11c and Figure 5.11d. The three red hyperplanes (lines) H^l are all loose hyperplanes since $\dim S^{H^l} = 0 < 1$.

For any $x \in (l, u]$ we define $H(x) := \operatorname{aff}\{(x, f(x)), (l, f(l))\}$ which is a nonvertical hyperplane. Consider any point $(x^0, f(x^0)) \neq (l, f(l))$ on the graph. The hyperplane $H(x^0)$ is below the point set $\{(x^0, f(x^0))\}$. For a fixed given $\bar{x} \in (l, u)$, $H(x^0)$ has exactly one intersection point with the vertical line $\{(x, y) \mid x = \bar{x}\}$. Thus, there exists a continuous function $g : (l, u] \to \mathbb{R}$ such that for any $x^0 \in (l, u]$, $(\bar{x}, g(x^0))$ is the intersection point of $H(x^0)$ and $\{(x, y) | x = \bar{x}\}$ (black points). Then for any $x^1, x^2 \in (l, u]$, $g(x^1) \leq g(x^2)$ implies that $H(x^1)$ is below $\{(x^2, f(x^2))\}$. For any $l' \in (l, u]$, there exists

$$x^* \in \arg\min_{x \in [l', u]} g(x).$$

It implies then $H(x^*)$ is below the graph of f over $[l', u]$.

Recall that f is a polynomial function. Since the second derivative $f^{(2)}$ is a continuous function, there exists an $\varepsilon > 0$ such that $f^{(2)} \leq 0$ or $f^{(2)} \geq 0$ for all $x \in [l, l+\varepsilon]$. It implies f is either a concave or a convex function over $[l, l+\varepsilon]$. After finding

$$x^* \in \arg \min_{x \in [l+\varepsilon, u]} g(x),$$

$H(x^*)$ is then below the graph of f over $[l+\varepsilon, u]$ as discussed above. We then have the following cases all based on $H(x^*)$:

- Case 1, f is concave over $[l, l+\varepsilon]$. An example is shown in Figure 5.11b. Note that the nonvertical hyperplane $\operatorname{aff}\{(l, f(l)), ((l+\varepsilon), f(l+\varepsilon))\}$ is below the graph of f over $[l, l+\varepsilon]$ and $H(x^*)$ is below $\operatorname{aff}\{(l, f(l)), ((l+\varepsilon), f(l+\varepsilon))\}$ over $[l, l+\varepsilon]$ by Lemma 5.18. It implies that $H(x^*)$ is below the graph of f over $[l, l+\varepsilon] \cup [l+\varepsilon, u]$, thus $H(x^*)$ is valid. The hyperplane $H(x^*)$ is tight since

$$1 \geq \dim S^{H(x^*)} \geq \dim \operatorname{aff}\{l, x^*\} = 1 \Rightarrow \dim S^{H(x^*)} = 1.$$

- Case $2a$, f is convex over $[l, l+\varepsilon]$ and the tangent plane $T(l)$ is below $H(x^*)$ over $[l, +\infty)$. An example is in Figure 5.11c. The tangent plane $T(l)$ is below the graph of f over $[l, l+\varepsilon]$ due to convexity. The tangent plane $T(l)$ is below the graph of f over $[l+\varepsilon, u]$ since $T(l)$ is below $H(x^*)$ over $[l, +\infty)$ and $H(x^*)$ is below the graph of f over $[l+\varepsilon, u]$. It implies then $T(l)$ is valid and $T(l)$ is tight since $T_h^{\max}(l) = T(l)$ has dimension 1.

- Case $2b$, f is convex over $[l, l+\varepsilon]$ and $H(x^*)$ is below $T(l)$ over $[l, +\infty)$. An example is in Figure 5.11d. Similar to the analysis in Case $2a$, $H(x^*)$ is valid. As discussed in Case 1, $H(x^*)$ is tight if it is valid.

We have only the three cases above since $T(l)$ is either below $H(x^*)$ or above $H(x^*)$ over $[l, +\infty)$ due to Lemma 5.18. □

Proof idea of Theorem 5.27. Now we go back to Theorem 5.27 with an arbitrary $n \in \mathbb{N}$. For any loose hyperplane H^l, we only need to prove that there always exists another valid hyperplane H^m with $\dim S^{H^m} > \dim S^{H^l}$ and $\pi_{\mathbf{x}}(H^l \cap \mathcal{S}) = X^{H^l} \subset X^{H^m} = \pi_{\mathbf{x}}(H^m \cap \mathcal{S})$. The corresponding tight hyperplane H^t can then be found iteratively.

To find such an H^m with $(H^m \cap \mathcal{S}) \supset (H^l \cap \mathcal{S})$ we first construct an affine set P of dimension $n-1$ with $S^{H^l} \subset P \subset H^l$. Note that for every valid hyperplane $H \supset P$ we have $\dim S^H \geq \dim S^{H^l}$. We can conclude the proof if we can find a valid hyperplane $H^m \supset P$ with $\dim S^{H^m} > \dim S^{H^l}$.

Similar to the proof for the special case $n = 1$, we split the domain X into two compact sets X^1 and X^2 with $X = X^1 \cup X^2$, $\pi_{\mathbf{x}}(P) \cap X^1 = \emptyset$ and $\pi_{\mathbf{x}}(P) \subset X^2$. Note that in the proof of the special case $n = 1$ above we have $P = \{(l, f(l))\}$, $X^1 = [l+\varepsilon, u]$ and $X^2 = [l, l+\varepsilon]$.

For every point $\mathbf{x}^0 \in X^1$, $H(\mathbf{x}^0) = \operatorname{aff}\{P, \{(\mathbf{x}^0, f(\mathbf{x}^0))\}\}$ is a nonvertical hyperplane. By solving an optimization problem we get $\mathbf{x}^* \in X^1$ such that $H(\mathbf{x}^*)$ is below the graph of f over X^1.

On the other hand, we want to find a hyperplane which is below the graph of f over X^2. Lemma 5.20 yields that a hyperplane is valid if and only if it is below the graph over all globally convex domain points. It is enough to find a hyperplane which is below the graph of f over $X^2 \cap \check{X}^g$. For that we find a hyperplane $H^B \supset P$ such that H^B is below $\mathcal{S}$ over $X^2 \cap \check{X}^g$.

At the end, we prove that one of $H(\mathbf{x}^*)$ and H^B is valid. Taking the valid one as H^m, we prove that it satisfies $\dim S^{H^m} > \dim S^{H^l}$ and $X^{H^l} \subset X^{H^m}$.

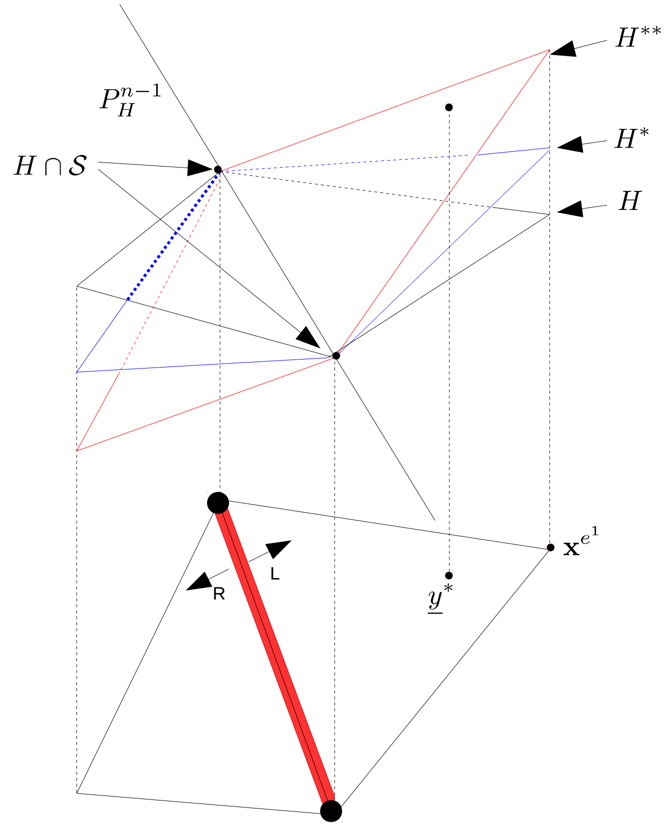


Figure 5.12: Example for Extendability

We are now ready to prove Theorem 5.27.

Proof (of Theorem 5.27). We only need to prove that for any loose hyperplane H^l there always exists a valid hyperplane H^m with $\dim S^{H^m} > \dim S^{H^l}$ and $\pi_{\mathbf{x}}(H^m \cap \mathcal{S}) \supset \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$. As we discussed in proof idea of Theorem 5.27 above, we first find a corresponding affine set P of dimension $n-1$ and split the domain X into X^1 and X^2.

Since $S^{H^l} \subset H^l$ is an affine set but not a hyperplane in $\mathbb{R}^{n+1}$, there exists an affine set $P^{n-1}_{H^l}$ of dimension $n-1$ with $S^{H^l} \subset P^{n-1}_{H^l} \subsetneq H^l$ as well as $(\pi_{\mathbf{x}}(P^{n-1}_{H^l}) \cap X^e) \subset \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$. For every point $(\mathbf{x}^0, z^0) \in \mathbb{R}^{n+1}$ with $\mathbf{x}^0 \notin \pi_{\mathbf{x}}(P^{n-1}_{H^l})$ and $z^0 \in \mathbb{R}$, $H(\mathbf{x}^0, z^0) = \operatorname{aff}\{P^{n-1}_{H^l}, \{(\mathbf{x}^0, z^0)\}\}$ is a nonvertical hyperplane in $\mathbb{R}^{n+1}$. Note that there are infinitely many such $P^{n-1}_{H^l}$ if $\dim S^{H^l} < n-1$. The set $X^e \setminus \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$ contains finitely many extreme points of X. The property $(\pi_{\mathbf{x}}(P^{n-1}_{H^l}) \cap X^e) \subset \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$ implies that $\pi_{\mathbf{x}}(P^{n-1}_{H^l}) \cap (X^e \setminus \pi_{\mathbf{x}}(H^l \cap \mathcal{S})) = \emptyset$. This makes sure that for any point $(\mathbf{x}^e, z)$ with $\mathbf{x}^e \in X^e \setminus \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$ and $z \in \mathbb{R}$, $H(\mathbf{x}^e, z)$ is a nonvertical hyperplane.

Now we want to split X into two compact sets X^1 and X^2 with $\pi_{\mathbf{x}}(P^{n-1}_{H^l}) \cap X^1 = \emptyset$, $\pi_{\mathbf{x}}(P^{n-1}_{H^l}) \subset X^2$ and $X = X^1 \cup X^2$. For that we first consider the following claim that we will prove later.

Claim a There exists a compact set X^M with $X \cap \pi_{\mathbf{x}}(P^{n-1}_{H^l}) \subset \operatorname{int} X^M$ such that a nonvertical hyperplane $H \supset P^{n-1}_{H^l}$ is below $\mathcal{S}$ over $X^M \cap \check{X}^g$ if H is below $T(\mathbf{x}^0)$ over X for every $\mathbf{x}^0 \in \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$.

Let $(\mathbf{x}^P, f(\mathbf{x}^P)) \in P^{n-1}_{H^l}$. Then there exists a normal vector $\mathbf{n}^P \in \mathbb{R}^n$ such that $\pi_{\mathbf{x}}(P^{n-1}_{H^l}) = \{\mathbf{x} \mid \mathbf{n}^P \cdot (\mathbf{x} - \mathbf{x}^P) = 0\}$. We define

$$X^L = \{\mathbf{x} \mid \mathbf{n}^P \cdot (\mathbf{x} - \mathbf{x}^P) \leq 0\} \text{ and } X^R = \{\mathbf{x} \mid \mathbf{n}^P \cdot (\mathbf{x} - \mathbf{x}^P) \geq 0\}$$

as well as

$$X^l = \{\mathbf{x} \mid \mathbf{n}^P \cdot (\mathbf{x} - \mathbf{x}^P) < 0\} \text{ and } X^r = \{\mathbf{x} \mid \mathbf{n}^P \cdot (\mathbf{x} - \mathbf{x}^P) > 0\}.$$

An example for $n = 2$ is shown in Figure 5.12.

Recall the definition of $X^e_{H^l}$ in Lemma 5.26. We have either $X^e \setminus (X^L \cup X^e_{H^l}) \neq \emptyset$ or $X^e \setminus (X^R \cup X^e_{H^l}) \neq \emptyset$. Due to symmetry, it suffices to consider the case $X^e \setminus (X^L \cup X^e_{H^l}) \neq \emptyset$. After that we set $X^1 = X^L \setminus \operatorname{int} X^M$ and $X^2 = X^R \cup X^M$. We have then $X = X^1 \cup X^2$, $\pi_{\mathbf{x}}(P^{n-1}_{H^l}) \cap X^1 = \emptyset$ and $\pi_{\mathbf{x}}(P^{n-1}_{H^l}) \subset X^2$. Note that X^1 and X^2 are both compact.

Claim b There exists a hyperplane $H^* \supset P^{n-1}_{H^l}$ that is valid over X^M. In addition, if H^* is valid over X then we have $\dim S^{H^*} > \dim S^{H^l}$.

Claim c There exists a hyperplane $H^{**} \supset P^{n-1}_{H^l}$ that is valid over X^1. In addition, if H^{**} is valid over X then we have $\dim S^{H^{**}} > \dim S^{H^l}$.

Claim d Either H^* or H^{**} is valid. The valid one is the hyperplane H^m we are looking for.

Now we prove these claims in a sequence, starting with Claim a. According to Lemma 5.21, every valid hyperplane H through $(\mathbf{x}^0, f(\mathbf{x}^0))$ is below $T(\mathbf{x}^0)$ over X. Let g_H denote the affine function such that $H = \{(\mathbf{x}, z) \mid z = g_H(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n\}$. Consider $\pi_{\mathbf{x}}(H^l \cap \mathcal{S})$ which is a compact set since $H^l \cap \mathcal{S} = \{(\mathbf{x}, z) \mid z = g_{H^l}(\mathbf{x}), \mathbf{x} \in X\} \cap \mathcal{S}$ is a compact set. It it clear that $\pi_{\mathbf{x}}(H^l \cap \mathcal{S}) \subset \check{X}^g$. According to Lemma 5.21, there exists a function $g_\varepsilon : \pi_{\mathbf{x}}(H^l \cap \mathcal{S}) \to \mathbb{R}$ such that for any $\mathbf{x}^0 \in \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$, $T(\mathbf{x}^0)$ is below $\mathcal{S}$ over $\check{X}^g \cap B_{\varepsilon_{\mathbf{x}^0}}(\mathbf{x}^0)$, where we abbreviate $\varepsilon_{\mathbf{x}^0} = g_\varepsilon(\mathbf{x}^0)$. The function g_ε possesses a minimum over $\pi_{\mathbf{x}}(H^l \cap \mathcal{S})$. Let $\varepsilon^{\min}$ be the minimum. It implies that that for any $\mathbf{x}^0 \in \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$, $T(\mathbf{x}^0)$ is below $\mathcal{S}$ over $\check{X}^g \cap B_{\varepsilon^{\min}}(\mathbf{x}^0)$. Define the set

$$B_{\varepsilon^{\min}}(\pi_{\mathbf{x}}(H^l \cap \mathcal{S})) = \bigcup_{\mathbf{x}^0 \in \pi_{\mathbf{x}}(H^l \cap \mathcal{S})} B_{\varepsilon^{\min}}(\mathbf{x}^0)$$

which is a compact set with $\pi_{\mathbf{x}}(H^l \cap \mathcal{S}) \subset \operatorname{relint} B_{\varepsilon^{\min}}(\pi_{\mathbf{x}}(H^l \cap \mathcal{S}))$. Further consider the compact set

$$\left(X \cap \pi_{\mathbf{x}}(P^{n-1}_{H^l})\right) \setminus \operatorname{relint} B_{\varepsilon^{\min}}(\pi_{\mathbf{x}}(H^l \cap \mathcal{S})).$$

Let g_P be the affine function such that $P^{n-1}_{H^l} = \{(\mathbf{x}, z) \mid z = g_P(\mathbf{x}), \mathbf{x} \in \pi_{\mathbf{x}}(P^{n-1}_{H^l})\}$. The validity of H^l implies that $g_P(\mathbf{x}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in X \cap \pi_{\mathbf{x}}(P^{n-1}_{H^l})$, $g_P(\mathbf{x}) = f(\mathbf{x})$ for all $\mathbf{x} \in \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$ and $g_P(\mathbf{x}) < f(\mathbf{x})$ for all $\mathbf{x} \in \left(X \cap \pi_{\mathbf{x}}(P^{n-1}_{H^l})\right) \setminus \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$. Thus, $g_P(\mathbf{x}) < f(\mathbf{x})$ for all $\mathbf{x} \in \left(X \cap \pi_{\mathbf{x}}(P^{n-1}_{H^l})\right) \setminus \operatorname{relint} B_{\varepsilon^{\min}}(\pi_{\mathbf{x}}(H^l \cap \mathcal{S}))$. For every valid hyperplane $H \supset P^{n-1}_{H^l}$, there exists a function

$$g^M : \left(X \cap \pi_{\mathbf{x}}(P^{n-1}_{H^l})\right) \setminus \operatorname{relint} B_{\varepsilon^{\min}}(\pi_{\mathbf{x}}(H^l \cap \mathcal{S})) \to \mathbb{R}$$

such that $g_H(\mathbf{x}) < f(\mathbf{x})$ for all $\mathbf{x} \in B_{\varepsilon^M_{\mathbf{x}^0}}(\mathbf{x}^0)$, where $\varepsilon^M_{\mathbf{x}^0} = g^M(\mathbf{x}^0)$. Note that g^M possesses a minimum $\varepsilon^{\min}_M$ over $\left(X \cap \pi_{\mathbf{x}}(P^{n-1}_{H^l})\right) \setminus \operatorname{relint} B_{\varepsilon^{\min}}(\pi_{\mathbf{x}}(H^l \cap \mathcal{S}))$. After that we define

$$X^M = \left(\bigcup_{\mathbf{x}^0 \in \left(X \cap \pi_{\mathbf{x}}(P^{n-1}_{H^l})\right) \setminus \operatorname{relint} B_{\varepsilon^{\min}}(\pi_{\mathbf{x}}(H^l \cap \mathcal{S}))} B_{\varepsilon^{\min}_M}(\mathbf{x}^0) \right) \cup B_{\varepsilon^{\min}}(\pi_{\mathbf{x}}(H^l \cap \mathcal{S})).$$

Analyzing all points in X^M shows Claim a.

Now we prove Claim b, c and d successively. We first construct a hyperplane $H^* \supset P^{n-1}_{H^l}$ with the property that if H^* is valid over X then there exists $\mathbf{x}^{e^*}$ such that $\mathbf{x}^{e^*} \in (X^e \cap X^1) \setminus X^e_{H^l}$ and $\mathbf{x}^{e^*} \in X^e_{H^*}$. This implies $\dim S^{H^*} > \dim S^{H^l}$. If H^* is valid we are done. Otherwise we finally prove that we are always able to find a hyperplane $H^{**} \supset P^{n-1}_{H^l}$ which is valid and there exists $\mathbf{x}^{**} \in X^1$ such that $(\mathbf{x}^{**}, f(\mathbf{x}^{**})) \in H^{**}$. This implies also $\dim S^{H^*} > \dim S^{H^l}$.

In the following, we first find an $H^* \supset P^{n-1}_{H^l}$ which is below $\mathcal{S}$ over $X^M \cap \check{X}^g$. According to Lemma 5.21 above, we only need to find an H^* which is below $T(\mathbf{x}^0)$ over X for every $\mathbf{x}^0 \in \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$. Hyperplane H is below $T(\mathbf{x}^0)$ over X for every $\mathbf{x}^0 \in \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$ if and only if H is below $T(\mathbf{x}^0)$ over X^e for every $\mathbf{x}^0 \in \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$. This is equivalent to the statement that for every $\mathbf{x}^e \in X^e$ it holds $g_H(\mathbf{x}^e) \leq \nabla f(\mathbf{x}^0) \cdot (\mathbf{x}^{e^i} - \mathbf{x}^0) + f(\mathbf{x}^0)$ for all $\mathbf{x}^0 \in \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$. Recall the definition of X^e_H defined in Lemma 5.26 with $\dim S^H = \dim \operatorname{aff}\{X^e_H\}$ for a valid hyperplane H. Any $\mathbf{x}^e \in X^e$ satisfies $\mathbf{x}^e \in X^e_H$ if and only if there exists $\mathbf{x}^0 \in \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$ such that $g_H(\mathbf{x}^e) = \nabla f(\mathbf{x}^0) \cdot (\mathbf{x}^e - \mathbf{x}^0) + f(\mathbf{x}^0)$. We only need to find an H^* such that for every $\mathbf{x}^e \in X^e$ $g_{H^*}(\mathbf{x}^e) \leq \nabla f(\mathbf{x}^0) \cdot (\mathbf{x}^{e^i} - \mathbf{x}^0) + f(\mathbf{x}^0)$ holds for all $\mathbf{x}^0 \in \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$ and there exists $\mathbf{x}^{e^*} \in X^e \setminus X^e_{H^l}$ such that there exists $\mathbf{x}^0 \in \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$ such that $g_{H^*}(\mathbf{x}^e) = \nabla f(\mathbf{x}^0) \cdot (\mathbf{x}^e - \mathbf{x}^0) + f(\mathbf{x}^0)$. Due to Lemma 5.26, if H^* is valid, we have then $\dim S^{H^*} = \dim \operatorname{aff}\{X^e_{H^*}\} > \dim \operatorname{aff}\{X^e_{H^l}\} = \dim S^{H^l}$ since $X^e_{H^*} \supset X^e_{H^l} \cup \{\mathbf{x}^{e^*}\}$.

Note that $X^e \setminus (\{X^e_{H^l}\}) \cup \pi_{\mathbf{x}}(P^{n-1}_{H^l})) \neq \emptyset$. Thus we assume $(X^l \cap X^e) \setminus X^e_{H^l} \neq \emptyset$ without loss of generality. For every $\mathbf{x}^{e^i} \in X^l \cap X^e$ and every $\mathbf{x}^0 \in X$, $(\mathbf{x}^{e^i}, z^{e^i})$ is a point on $T(\mathbf{x}^0)$ with $z^{e^i} = \nabla f(\mathbf{x}^0) \cdot (\mathbf{x}^{e^i} - \mathbf{x}^0) + f(\mathbf{x}^0)$. The affine set $H^i_0 = \operatorname{aff}\{P^{n-1}_H, \{(\mathbf{x}^{e^i}, z^{e^i})\}\}$ is then a hyperplane denoted by $H^i_0 = \{(\mathbf{x}, z) \mid z = g_{H^i_0}(\mathbf{x})\}$ that depends on $\mathbf{x}^{e^i} \in (X^l \cap X^e) \setminus \{X^e_{H^l}\}$ and $\mathbf{x}^0 \in X$. For a fixed given $\mathbf{y}^* \in X^l$ and $\mathbf{x}^{e^i} \in (X^l \cap X^e) \setminus \{X^e_{H^l}\}$, there exists a continuous function

$$g^{e^i}_f : \pi_{\mathbf{x}}(P^{n-1}_H) \cap X \to \mathbb{R} \text{ with } g_{H^i_0}(\mathbf{y}^*) = g^{e^i}_f(\mathbf{x}^0).$$

Since $\pi_{\mathbf{x}}(H^l \cap \mathcal{S}) \subset \pi_{\mathbf{x}}(P^{n-1}_H) \cap X$ is a compact set, the extreme value theorem of Weierstrass yields that $g^{e^i}_f$ possesses a minimum over $\pi_{\mathbf{x}}(H \cap \mathcal{S})$. We set

$$z^{e^i}_f = \min_{\mathbf{x} \in \pi_{\mathbf{x}}(H^l \cap \mathcal{S})} g^{e^i}_f(\mathbf{x}).$$

Since we have finitely many $\mathbf{x}^{e^i} \in (X^l \cap X^e) \setminus X^e_{H^l}$, we further set

$$z^{\min}_f = \min_{\mathbf{x}^{e^i} \in (X^l \cap X^e) \setminus \{X^e_{H^l}\}} z^{e^i}_f(\mathbf{x}^{e^i})$$

and

$$\mathbf{x}^{e^i}_{\min} \in \arg \min_{\mathbf{x}^{e^i} \in (X^l \cap X^e) \setminus \{X^e_{H^l}\}} z^{e^i}_f(\mathbf{x}^{e^i})$$

as well as $H^* = \operatorname{aff}\{P^{n-1}_{H^l}, \{(\mathbf{x}^{e^i}_{\min}, z^{\min}_f)\}\}$. We then have that H^* is above H^l over X^L and below H^l over X^R. Furthermore, for every $\mathbf{x}^0 \in \pi_{\mathbf{x}}(H \cap \mathcal{S})$, H^* is below $T(\mathbf{x}^0)$ over X since H^* is below $(\mathbf{x}^{e^i}, g_{T(\mathbf{x}^0)}(\mathbf{x}^{e^i}))$ for every $\mathbf{x}^{e^i} \in X^e$. If H^* is valid, then we have $X^e_{H^*} \supset X^e_{H^l} \cup \{\mathbf{x}^{e^i}_{\min}\}$ with $\mathbf{x}^{e^i}_{\min} \notin X^e_{H^l}$ which implies $\dim S^{H^*} > \dim S^{H^l}$ and we are done.

Otherwise H^* is not valid. Then there exists $\mathbf{x}^f \in X^1$ such that $g_{H^*}(\mathbf{x}^f) > f(\mathbf{x}^f)$. Note that for every $\mathbf{y} \in X^1$, $H(\mathbf{y}) = \operatorname{aff}\{P^{n-1}_{H^l}, \{(\mathbf{y}, f(\mathbf{y}))\}\}$ is a hyperplane in $\mathbb{R}^{n+1}$ written as $\{(\mathbf{x}, z) \mid g_{H(\mathbf{y})}(\mathbf{x})\}$, where $g_{H(\mathbf{y})}$ is an affine function depending on $\mathbf{y}$. Function $g(\mathbf{y}) := g_{H(\mathbf{y})}(\mathbf{x}^f)$ is then a continuous function of $\mathbf{y}$. Since X^1 is a compact set, $g(\mathbf{y})$ possesses a minimum $g(\mathbf{y}^{**})$ over X^1. The corresponding hyperplane $H^{**} = \operatorname{aff}\{P^{n-1}_{H^l}, \{(\mathbf{y}^{**}, f(\mathbf{y}^{**}))\}\}$ is then below $\mathcal{S}$ over X^1 since for every other $\mathbf{y} \in X^1$, H^{**} is below $\operatorname{aff}\{P^{n-1}_{H^l}, \{(\mathbf{y}, f(\mathbf{y}))\}\}$

over X^1 which implies $(\mathbf{y}, f(\mathbf{y})) \in \operatorname{aff}\{P^{n-1}_{H^l}, \{(\mathbf{y}, f(\mathbf{y}))\}\}$ is above H^{**}. Second, note that $g_{H^*}(\mathbf{x}^f) > f(\mathbf{x}^f) \geq g_{H^{**}}(\mathbf{x}^f)$ implies H^{**} is below H^* over X^L. Since H^* is valid over $X^M \cap \check{X}^g$, H^{**} is valid over $X^L \cap X^M \cap \check{X}^g$. Finally, with Lemma 5.18, H^{**} is above H^l over X^L since

$$g_{H^{**}}(\mathbf{y}^{**}) = f(\mathbf{y}^{**}) > g_{H^l}(\mathbf{y}^{**}).$$

Thus H^{**} is below H^l over X^R. Since H^l is valid, H^{**} is valid over X^R. As a consequence, H^{**} is valid since H^{**} is valid over $X^1 \cup (X^L \cap X^M \cap \check{X}^g) \cup X^R \supset \check{X}^g$.

We discussed above that if H^* is valid then $\dim S^{H^*} > \dim S^{H^l}$. Otherwise H^{**} is valid, $(\mathbf{y}^{**}, f(\mathbf{y}^{**})) \in H^{**} \cap \mathcal{S}$ and $(\mathbf{y}^{**}, f(\mathbf{y}^{**})) \notin S^{H^l}$ together imply that $\dim S^{H^{**}} > \dim S^{H^l}$. In both cases we set H^m as the valid hyperplane and we have $\dim S^{H^m} > \dim S^{H^l}$ and $\pi_{\mathbf{x}}(H^m \cap \mathcal{S}) \supset \pi_{\mathbf{x}}(H^l \cap \mathcal{S})$. □

This theorem means that every loose hyperplane can be extended to be a tight hyperplane with the intersecting points preserved. Recall that due to Theorem 5.1, the convex hull of $\mathcal{S}$ is the intersection of all closed halfspaces containing $\mathcal{S}$. In the following section we show that we need only those halfspaces whose corresponding hyperplanes are tight.

5.3.6 Convex hull that only consists of tight halfspaces

Theorem 5.28 (Convex hull of the graph of polynomial functions)
Let $f \in \mathbb{R}[\mathbf{x}]$ be a polynomial function and $\mathcal{S}$ the graph of f over X defined as

$$\mathcal{S} := \{(\mathbf{x}, z) \mid z = f(\mathbf{x}),\ \mathbf{x} \in X\} \subset \mathbb{R}^{n+1}.$$

The downward closed part $\mathrm{co\check{n}v}(\mathcal{S}) = conv(\{(\mathbf{x}, z) \mid z \geq f(\mathbf{x}), \mathbf{x} \in X\})$ of the convex hull $conv(\mathcal{S})$ of $\mathcal{S}$ can be represented as

$$\mathrm{co\check{n}v}(\mathcal{S}) = \left(\bigcap_{\substack{H \text{ is downward valid} \\ \text{and tight}}} \check{H} \right) \cap \left(\{(\mathbf{x}, z) \mid \mathbf{x} \in X\}\right). \tag{5.15}$$

Analogously, we define the upward closed part $\mathrm{co\hat{n}v}(\mathcal{S}) = conv(\{(\mathbf{x}, z) \mid z \leq f(\mathbf{x}), \mathbf{x} \in X\})$. Then

$$conv(\mathcal{S}) = \mathrm{co\check{n}v}(\mathcal{S}) \cap \mathrm{co\hat{n}v}(\mathcal{S}). \tag{5.16}$$

Before we prove Theorem 5.28 we need the following lemma.

Lemma 5.29
For every point $\mathbf{x}^0 \in X$, there exists a downward valid hyperplane H such that

$$\mathbf{x}^0 \in conv\left(\pi_{\mathbf{x}}(H \cap \mathcal{S})\right).$$

Proof. First we introduce a few definitions and notes from Jach et al. [JMW08]. The convex envelope of f over X can be defined as

$$\begin{aligned}\mathrm{vex}_X[f](\mathbf{x}) = \sup \{\, \eta(\mathbf{x}) \mid \eta : X \to \mathbb{R} \text{ with}\\ \eta(\mathbf{x}) \le f(\mathbf{x}) \text{ for all } \mathbf{x} \in X, \text{ and } \eta \text{ convex} \,\}.\end{aligned}$$

For any $\mathbf{x}^0 \in X$, the value $\mathrm{vex}_X[f](\mathbf{x}^0)$ can be represented by

$$\mathrm{vex}_X[f](\mathbf{x}^0) = \min \{\mu \mid (\mathbf{x}^0, \mu) \in \mathrm{conv}(\mathcal{S})\}.$$

The value $\mu^0 = \mathrm{vex}_X[f](\mathbf{x}^0)$ can be obtained by solving the nonlinear optimization problem $(\mathrm{OP}_{\mathbf{x}^0})$

$$\begin{aligned}\mathrm{vex}_X[f](\mathbf{x}^0) &= \min \mu\\ \text{s.t.} \sum_{k=1}^{n+1} \lambda_k \mathbf{x}^k &= \mathbf{x}^0,\\ \sum_{k=1}^{n+1} \lambda_k f(\mathbf{x}^k) &= \mu,\\ \sum_{k=1}^{n+1} \lambda_k &= 1,\\ \lambda_k &\ge 0, k = 1, \dots, n+1,\\ \mathbf{x}^k &\in X, k = 1, \dots, n+1.\end{aligned}$$

Without loss of generality, this optimization problem has an optimal solution with $\lambda_k > 0$ for $k = 1, \dots, t'$ and $\lambda_k = 0$ for $k = t'+1, \dots, n+1$, where t' is an integer with $1 \le t' \le n+1$. Then

$$\mathbf{x}^0 \in \mathrm{conv}\left(\{\mathbf{x}^1, \dots, \mathbf{x}^{t'}\}\right) =: X_{\mathbf{x}^0}.$$

Consider the affine set

$$P = \mathrm{aff}\{(\mathbf{x}^1, f(\mathbf{x}^1)), \dots, (\mathbf{x}^{t'}, f(\mathbf{x}^{t'}))\}.$$

If P is a hyperplane, we prove that P is valid. If P is not valid, then there exists $(\mathbf{x}^b, f(\mathbf{x}^b))$ which is below P. Consider first the case $\mathbf{x}^b \in X_{\mathbf{x}^0}$, then there exist $\lambda'_k \ge 0$ for $k = 1, \dots, t'$ such that

$$\mathbf{x}^b = \sum_{k=1}^{t'} \lambda'_k \mathbf{x}^k$$

with

$$1 = \sum_{k=1}^{t'} \lambda'_k$$

and

$$f(\mathbf{x}^b) < \sum_{k=1}^{t'} \lambda'_k f(\mathbf{x}^k),$$

since $(\mathbf{x}^b, f(\mathbf{x}^b))$ is below P. We set further

$$\beta := \min_{\substack{\lambda'_k \neq 0 \\ k \in \{1,\dots,t'\}}} \frac{\lambda_k}{\lambda'_k}.$$

Since the λ_k are all positive and at least one of λ'_k is nonzero, we have $\beta > 0$. Without loss of generality, we have $\beta = \lambda_1/\lambda'_1$, then it holds $\beta\lambda'_1 = \lambda_1$ and $\beta\lambda'_k \leq \lambda_k$ for any $k \in \{2, \dots, t'\}$. Note that

$$\begin{aligned}
\mathbf{x}^0 &= \sum_{k=1}^{t'} \lambda_k \mathbf{x}^k \\
&= \lambda_1 \mathbf{x}^1 + \sum_{k=2}^{t'} \lambda_k \mathbf{x}^k \\
&= \beta\lambda'_1 \mathbf{x}^1 + \sum_{k=2}^{t'} (\lambda_k - \beta\lambda'_k + \beta\lambda'_k)\mathbf{x}^k \\
&= \beta \sum_{k=1}^{t'} \lambda'_k \mathbf{x}^k + \sum_{k=2}^{t'} (\underbrace{\lambda_k - \beta\lambda'_k}_{\geq 0})\mathbf{x}^k \\
&= \beta \mathbf{x}^b + \sum_{k=2}^{t'} (\underbrace{\lambda_k - \beta\lambda'_k}_{=:\lambda''_k})\mathbf{x}^k.
\end{aligned}$$

Hence $(\mathrm{OP}_{\mathbf{x}^0})$ has a new solution using $\mathbf{x}^b, \mathbf{x}^2, \dots, \mathbf{x}^{t'}$ with coefficients $\beta, \lambda''_2, \dots, \lambda''_{t'}$ and objective

$$u' = \beta f(\mathbf{x}^b) + \sum_{k=2}^{t'} (\lambda_k - \beta\lambda'_k) f(\mathbf{x}^k).$$

We compare now u' and u by computing

$$\begin{aligned} u' - u &= \beta f(\mathbf{x}^b) + \sum_{k=2}^{t'} (\lambda_k - \beta\lambda'_k) f(\mathbf{x}^k) - \sum_{k=1}^{t'} \lambda_k f(\mathbf{x}^k) + \underbrace{(\lambda_1 - \beta\lambda'_1)}_{=0} f(\mathbf{x}^1) \\ &= \beta f(\mathbf{x}^b) + \sum_{k=1}^{t'} (\lambda_k - \beta\lambda'_k) f(\mathbf{x}^k) - \sum_{k=1}^{t'} \lambda_k f(\mathbf{x}^k) \\ &= \beta (f(\mathbf{x}^b) - \sum_{k=1}^{t'} \lambda'_k f(\mathbf{x}^k)) \\ &< 0. \end{aligned}$$

It means that there exists a new solution that decreases the current optimal value strictly which is a contradiction. Hence P is valid. Analogously, for the case $\mathbf{x}^b \notin X_{\mathbf{x}^0}$ we can also prove that P is valid. With the same proof approach it implies that every $\mathbf{x}^i$ is a globally convex domain point.

Consider the case P is not a hyperplane with $\dim(P) \leq n-1$. Now we prove that there exists a valid H satisfying $P \subset H$. Due to Corollary 5.13, for $i \in \{1, \dots, t'\}$, any valid hyperplane H which contains $(\mathbf{x}_i, f(\mathbf{x}_i))$, satisfies also $T_{I_{\mathbf{x}^i}}(\mathbf{x}^i) \subset H$. We only need to consider the case that the points $\mathbf{x}^1, \dots, \mathbf{x}^{t'}$ are all boundary domain points. Otherwise for any $\mathbf{x}^i \in \operatorname{int} X$, $H = T(\mathbf{x}^i)$ is the hyperplane we are looking for. In this case we only have two sub-cases to prove, $\mathbf{x}^0 \in \partial X$ or $\mathbf{x}^0 \in \operatorname{int} X$. Both cases imply that $X_{\mathbf{x}^0}$ is a face of X.

First consider the case $\mathbf{x}^0 \in \partial X$. Due to Theorem 5.12, by setting $F_{\mathbf{x}^0} = X_{\mathbf{x}^0}$ there, we find a valid hyperplane H with $F_{\mathbf{x}^0} \subset H$. This implies the result.

Second, the case $\mathbf{x}^0 \in \operatorname{int} X$ remains to be proved. This implies that $\mathbf{x}^0 \in \operatorname{relint}(X_{\mathbf{x}^0})$.

The definition of valid hyperplanes can be easily extended to "valid affine spaces". An affine space $P \in \mathbb{R}^{n+1}$ of dimension $n' < n$ is said to be valid if P is below $\mathcal{S}$ over $X \cap \pi_{\mathbf{x}}(P)$.

It is clear that $P^{'}$ is a valid affine set. Otherwise we can improve the found optimum. Now we only need to prove that there exists a valid plane $P^{''}$ with $P^{'} \subset P^{''}$ and $\dim(P^{''}) = n' + 1$ since then we can find the valid hyperplane H with required conditions recursively.

Consider the set $X^{'}_{\mathbf{x}^0} = \pi_{\mathbf{x}}(P^{'})$ with $\dim(X^{'}_{\mathbf{x}^0}) = n' - 1$. There exists $n - n'$ affine sets $H^{'}_1, \dots, H^{'}_{n-n'}$ such that it can be represented as

$$X^{'}_{\mathbf{x}^0} = \underbrace{H^{'}_1}_{=\{\mathbf{x}\in\mathbb{R}^n | g_1(\mathbf{x})=0\}} \bigcap \dots \bigcap H^{'}_{n-n'},$$

where there exists a function g_1 such that $H^{'}_1 = \{\mathbf{x} \in \mathbb{R}^n \mid g_1(\mathbf{x}) = 0\}$.

Similarly, we define

$$X^L_{\mathbf{x}^0} = \{\mathbf{x} \in \mathbb{R}^n \mid g_1(\mathbf{x}) \leq 0\} \bigcap H^{'}_1 \bigcap \dots \bigcap H^{'}_{n-n'}$$

and

$$X^R_{\mathbf{x}^0} = \{\mathbf{x} \in \mathbb{R}^n \mid g_1(\mathbf{x}) \geq 0\} \bigcap H^{'}_1 \bigcap \ldots \bigcap H^{'}_{n-n'}$$

as well as

$$X^{'}_L = X \bigcap X^L_{\mathbf{x}^0} \text{ and } X^{'}_R = X \bigcap X^R_{\mathbf{x}^0}.$$

Note that $\mathbf{x}^0$ is now a boundary point for $X^{'}_L$ and for $X^{'}_R$. Due to Theorem 5.12, there exist hyperplanes H^L and H^R such that

- $P^{'} \subset (H^L \cap H^R)$
- H^L is below $\mathcal{S}$ over $X^{'}_L$, i.e., H^L is valid over $X^{'}_L$,
- any $H^{L'} \supset P^{'}$ which is above H^L over $X^{'}_L$ is not valid over $X^{'}_L$,
- H^R is below $\mathcal{S}$ over $X^{'}_R$,
- any $H^{R'} \supset P^{'}$ which is above H^R over $X^{'}_R$ is not valid over $X^{'}_R$.

Let $\mathbf{x}^L \in \operatorname{relint} X^{'}_L$ and $\mathbf{x}^R \in \operatorname{relint} X^{'}_R$ be two arbitrary points with $(\mathbf{x}^L, z^L) \in H^L$ and $(\mathbf{x}^R, z^R) \in H^R$ points on the hyperplanes, respectively. We further denote

$$P^{''}_L = \operatorname{aff}\left\{P^{'}, \{(\mathbf{x}^L, z^L)\}\right\} \text{ and } P^{''}_R = \operatorname{aff}\left\{P^{'}, \{(\mathbf{x}^R, z^R)\}\right\}.$$

A graphic example is shown in Figure 5.13.

According to Lemma 5.18, by comparing $P^{''}_L$ and $P^{''}_R$ we have only two cases:

1. The affine set $P^{''}_L$ is below $P^{''}_R$ over $X^{'}_R$. This is equivalent to $P^{''}_R$ is below $P^{''}_L$ over $X^{'}_L$.
2. The affine set $P^{''}_L$ is above $P^{''}_R$ over $X^{'}_R$. This is equivalent to $P^{''}_R$ is above $P^{''}_L$ over $X^{'}_L$.

In case 1, both $P^{''}_L$ and $P^{''}_R$ are valid affine sets and we are finished.

To conclude the proof we show that case 2 is not possible. In case 2, $P^{''}_R$ is above $P^{''}_L$ over $X^{'}_L$. Take any affine set $P^{''}_M$ with $\dim(P^{''}_M) = n' + 1$, $P^{'} \subset P^{''}_M$ and $P^{''}_R$ being above $P^{''}_M$ over $X^{'}_L$ as well as $P^{''}_M$ being above $P^{''}_L$ over $X^{'}_L$. Since $P^{''}_M$ is neither valid in $X^{'}_L$ nor valid in $X^{'}_L$, there exist intersection points $\mathbf{x}^{l'} \in \operatorname{relint} X^{'}_L$ with $(\mathbf{x}^{l'}, f(\mathbf{x}^{l'})) \in P^{''}_M$ and $\mathbf{x}^{r'} \in \operatorname{relint} X^{'}_R$ with $(\mathbf{x}^{r'}, f(\mathbf{x}^{r'})) \in P^{''}_M$.

Let g_M be a function denoting $P^{''}_M = \{\mathbf{x} \mid g_M(\mathbf{x}) = 0\}$. There exist $\mathbf{x}^l \in \operatorname{relint} X^{'}_L \cap B_{\varepsilon_L}(\mathbf{x}^{l'})$ and $\mathbf{x}^r \in \operatorname{relint} X^{'}_R \cap B_{\varepsilon_R}(\mathbf{x}^{r'})$, where $B_{\varepsilon_L}(\mathbf{x}^{l'})$ and $B_{\varepsilon_R}(\mathbf{x}^{r'})$ are neighborhoods of $\mathbf{x}^{l'}$ and $\mathbf{x}^{r'}$ for $\varepsilon_L > 0$ and $\varepsilon_R > 0$, respectively, such that

- $f(\mathbf{x}^l) < g_M(\mathbf{x}^l)$, i.e., point $(\mathbf{x}^l, f(\mathbf{x}^l))$ is below $P^{''}_M$,
- $f(\mathbf{x}^r) < g_M(\mathbf{x}^r)$, i.e., point $(\mathbf{x}^r, f(\mathbf{x}^r))$ is below $P^{''}_M$.

See the graphic example in Figure 5.13 again.

Recall that μ^0 is the optimal value of $\mathrm{OP}_{\mathbf{x}^0}$, then we have $f(\mathbf{x}^0) > \mu^0$. Consider the new polytope defined by

$$S^0 = \operatorname{conv}\left(\{(\mathbf{x}^0, f(\mathbf{x}^0)), (\mathbf{x}^1, f(\mathbf{x}^1)), \dots, (\mathbf{x}^{t'}, f(\mathbf{x}^{t'}), (\mathbf{x}^l, f(\mathbf{x}^l)), (\mathbf{x}^r, f(\mathbf{x}^r))\}\right).$$

It is clear that $S^0 \subset \operatorname{conv}(\mathcal{S})$ and $(\mathbf{x}^0, \mu^0) \in \operatorname{relint} S^0$. Consider the LP

$$\min\left\{\mu \mid (\mathbf{x}, \mu) \in S^0, \mathbf{x} = \mathbf{x}^0\right\},$$

the optimal objective function value μ^1 fulfills $\mu^1 < \mu^0$ which is a contradiction to the optimum of $\mathrm{OP}_{\mathbf{x}^0}$ since $S^0 \subset \operatorname{conv}\mathcal{S}$. Thus either P''_L or P''_R is valid. □

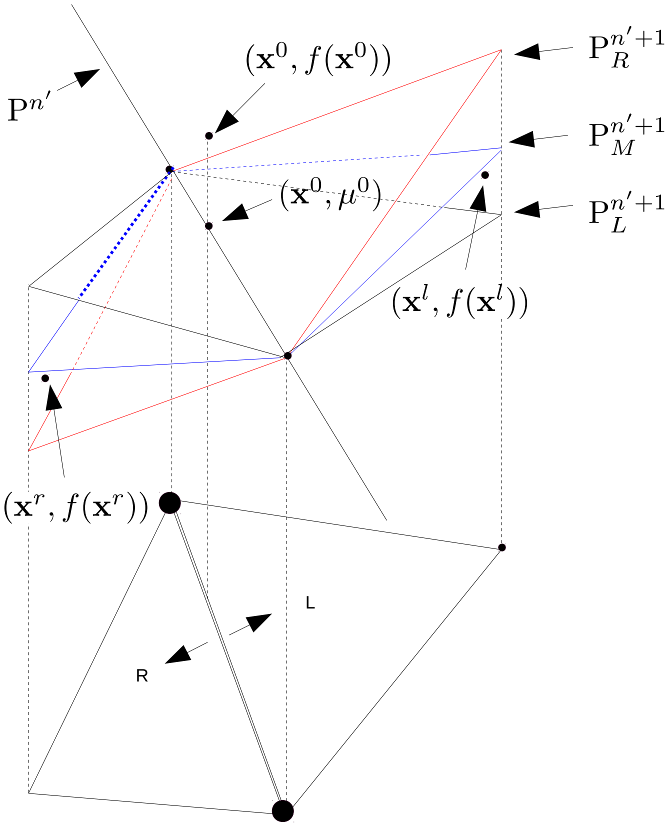


Figure 5.13: Example for the proof of Lemma 5.29

Proof (of Theorem 5.28). We show that the following sets are equivalent:

$$\begin{aligned}\check{\operatorname{conv}}(\mathcal{S}) &= \operatorname{conv}(\{(\mathbf{x}, z) \mid z \geq f(\mathbf{x}), \mathbf{x} \in X\}) \\ &= \left(\bigcap_{H \text{ is downward valid}} \check{H}\right) \cap (\{(\mathbf{x}, z) \mid \mathbf{x} \in X\}) =: S' \\ &= \left(\bigcap_{\substack{H \text{ is downward valid} \\ \text{and } H\cap\mathcal{S}\neq\emptyset}} \check{H}\right) \cap (\{(\mathbf{x}, z) \mid \mathbf{x} \in X\}) =: S'' \\ &= \left(\bigcap_{\substack{H \text{ is downward valid} \\ \text{and tight}}} \check{H}\right) \cap (\{(\mathbf{x}, z) \mid \mathbf{x} \in X\}) =: S'''.\end{aligned}$$

Theorem 5.1 and Lemma 5.2 then imply $\check{\operatorname{conv}}(\mathcal{S}) = S' = S''$.

Note that every tight hyperplane H fulfills $H \cap \mathcal{S} \neq \emptyset$. For every loose hyperplane H with $H \cap \mathcal{S} \neq \emptyset$, there exists a tight hyperplane which we proved in Theorem 5.27. Thus,

$$\emptyset \neq \{H \mid H \text{ is downward valid and tight}\} \subset \{H \mid H \text{ is downward valid and } H \cap \mathcal{S} \neq \emptyset\}$$

and

$$S'' \subset S'''.$$

For any $\mathbf{x}^0$ consider the optimization problem

$$\min\{z \mid (\mathbf{x}^0, z) \in S''\}. \qquad (\mathrm{OP}_1^{\mathbf{x}^0})$$

By Lemma 5.29 there exists H^0 which is downward valid with

$$\mathbf{x}^0 \in \operatorname{conv}\left(\pi_{\mathbf{x}}(H^0 \cap \mathcal{S})\right).$$

Let $(\mathbf{x}^0, z^0) \in H^0$. Then ($\mathrm{OP}_1^{\mathbf{x}^0}$) has the minimum z^0 since every downward valid hyperplane H is below $H^0 \cap \mathcal{S}$ and $(\mathbf{x}^0, z^0) \in \operatorname{conv}(H^0 \cap \mathcal{S})$.

Consider the optimization problem

$$\min\{z \mid (\mathbf{x}^0, z) \in S'''\}. \qquad (\mathrm{OP}_2^{\mathbf{x}^0})$$

If H^0 is tight, ($\mathrm{OP}_2^{\mathbf{x}^0}$) also has the minimum z^0. Otherwise Theorem 5.27 implies that there exists a tight and valid hyperplane H^1 such that

$$H^0 \cap \mathcal{S} \subset H^1.$$

This implies that $(\mathbf{x}^0, z^0) \in \operatorname{conv}(H^1 \cap \mathcal{S})$ and z^0 is the minimum of ($\mathrm{OP}_2^{\mathbf{x}^0}$) since every downward valid hyperplane H is below $H^1 \cap \mathcal{S}$.

Now we proved for every fixed $\mathbf{x}^0 \in X$ that ($\mathrm{OP}_1^{\mathbf{x}^0}$) and ($\mathrm{OP}_2^{\mathbf{x}^0}$) both have the same minimum. S'' and S''' are both formed by downward closed halfspaces, $S'' = S'''$ holds.

Equation (5.15) implies then (5.16) due to the symmetry. □

Let $\check{\mathcal{H}}^{\bowtie}$ be the set of all downward valid and tight hyperplanes and $\hat{\mathcal{H}}^{\bowtie}$ be the set of all upward valid and tight hyperplanes. Let $\check{\mathcal{H}}^{\diamond} \subset \check{\mathcal{H}}^{\bowtie}$ and $\hat{\mathcal{H}}^{\diamond} \subset \hat{\mathcal{H}}^{\bowtie}$ be two finite sets.

Corollary 5.30 (Polyhedral relaxation)
The set

$$S(\check{\mathcal{H}}^{\diamond}, \hat{\mathcal{H}}^{\diamond}) = \left(\bigcap_{\check{H} \in \check{\mathcal{H}}^{\diamond}} \check{H} \right) \cap \left(\bigcap_{\hat{H} \in \check{\mathcal{H}}^{\diamond}} \hat{H} \right)$$

is a polyhedral set which fulfills

$$conv(\mathcal{S}) \subset S(\check{\mathcal{H}}^{\diamond}, \hat{\mathcal{H}}^{\diamond}).$$

So far, we gave a description of the convex hull of the graph of polynomial functions. Recalling the definitions and theorems, our work focused mainly on theoretical point of view. Instead of obtaining algorithms to compute valid hyperplanes, we dealt with proof of existence. Indeed, algorithmically, it is very hard to verify if a given hyperplane is valid in a general dimension and for a general degree of polynomial functions.

In the next section, we concentrate on bivariate polynomial functions with a limited degree. Algorithms are developed to find tight hyperplanes. Computations show that these tight hyperplanes accelerate MINLP solving processes.

5.4 Bivariate polynomial functions: a case study

In this section we design algorithms to find finitely many tight valid hyperplanes for the graph of bivariate polynomial functions with degree up to 3. Every given bivariate polynomial function with degree up to 3 has the form

$$f(x,y) = \sum_{\substack{0 \le i,j \le 3 \\ 0 \le i+j \le 3}} a_{ij} x^i y^j, \tag{5.17}$$

where all $a_{ij} \in \mathbb{R}$ are constants and $X \subset \mathbb{R}^2$ is the domain which is a polytope. Then X is a convex polygon with $m \geq 3$ edges and vertices. Every edge is a line segment as well as a facet of X and every vertex is an extreme point of X.

Again, we only consider the downward closed part. Recall that $\check{X}^g$ is the set of all globally convex domain points and $\check{X}^l$ is the set of all locally convex domain points. Theoretically, for any $\mathbf{x}^0 \in \operatorname{int} X$ we need only to check if $T(\mathbf{x}^0)$ is valid. However, in practice, this is not easy even for f given as in (5.17). Instead of getting valid hyperplanes starting from interior domain points, we pay more attention to those boundary domain points.

Using the result from Section 5.3, the graph of the bivariate polynomial function f on a facet of X is isomorphic to the graph of a univariate polynomial function on a corresponding projected domain. We show later that finding $\check{X}^g$ for univariate polynomial functions with degree ≤ 3 is tractable. Thus we can easily find the set $\check{X}^g \cap \partial X$ for bivariate polynomial functions with degree ≤ 3. In the following we design algorithms which first compute a few hyperplanes that are below $\mathcal{S}$ over $\check{X}^g \cap \partial X$. For each of the hyperplanes which are below the boundary of $\mathcal{S}$, we solve a NLP globally either to verify if the hyperplane is valid or to find a valid hyperplane which is parallel to this hyperplane. These NLPs contain only two variables and can be globally solved by SCIP in less than one second.

Going back to our applications, all of these hyperplanes can be found in an offline way, i.e., before we start to solve the MINLPs. For every instance we need only to calculate these hyperplanes once. Every globally solved NLP above yields a tight valid hyperplane.

Remark 5.31
In this section we discuss hyperplanes and graphs of polynomial functions in $\mathbb{R}^3$. As before, we use (x, y, z) to denote a point in $\mathbb{R}^3$. Similar to Section 5.3, we use $\mathbf{x} = (x, y) \in \mathbb{R}^2$ to denote domain points and use e.g., $\mathbf{x}^0 = (x_0, y_0) \in \mathbb{R}^2$ to denote a certain domain point.

For a boundary point $(x, y) \in \partial X$ there exists at least one facet F_i of X with $(x, y) \in F_i$. Since F_i is a line segment, it must be contained in a line denoted by

$$\{(x, y) \mid a_i x + b_i y + c_i = 0\} =: \operatorname{aff}\{F_i\},$$

where $a_i, b_i, c_i \in \mathbb{R}$ are constants and at least one of a_i and b_i is nonzero. Without loss of generality we assume $b_i \neq 0$ (otherwise permute x and y) and set $b_i = 1$ (otherwise scale a_i, b_i and c_i). Facet F_i can be then be represented as

$$F_i = \{(x, y) \mid y = -a_i x - c_i, x \in [x_i^{\min}, x_i^{\max}]\},$$

where $x_i^{\min}, x_i^{\max} \in \mathbb{R}$ are constants with $x_i^{\min} < x_i^{\max}$. Recalling the definitions in Section 5.3 and using the same notations, we have the projection map

$$g_d : \operatorname{aff}\{F_i\} \to \mathbb{R}, (x, y) \mapsto x$$

and its inverse map

$$g_d^{-1} : \mathbb{R} \to \operatorname{aff}\{F_i\}, x \mapsto \begin{pmatrix} x \\ -a_i x - c_i \end{pmatrix}$$

as well as

$$f_i(x) = f(x, -a_i x - c_i) = \sum_{\substack{0 \leq i,j \leq 3 \\ 0 \leq i+j \leq 3}} a_{ij} x^i (-a_i x - c_i)^j = ax^3 + bx^2 + cx + d,$$

where a, b, c, d are constants depending on a_i, c_i and all a_{ij}. An example has been shown in Figure 5.4 and discussed in Section 5.3.

Corollary 5.32
A boundary domain point (x_0, y_0) on facet F_i of X is globally convex for $\mathcal{S}$ if and only if x_0 is globally convex for the graph of $f_i(x)$ over $[x_i^{\min}, x_i^{\max}]$.

Proof. The result is a special case of Theorem 5.12. □

Let $\check{X}_i^l \subset [x_i^{\min}, x_i^{\max}]$ denote the set of all locally convex domain points for the graph of $f_i(x)$ and $\check{X}_i^g \subset \check{X}_i^l$ the set of the globally convex domain points. Note that $\check{X}^g \cap F_i = g_d^{-1}(\check{X}_i^g)$. Thus, finding $\check{X}_i^g$ for every $i \in \{1, \ldots, m\}$ will find $\check{X}^g \cap \partial X$.

Lemma 5.33
The set of globally convex domain points $\check{X}_i^g \subset [x_i^{\min}, x_i^{\max}]$ has one of the four following forms

1. $\{x_i^{\min}, x_i^{\max}\}$,
2. $[x_i^{\min}, x_i^{\max}]$,
3. $[x_i^{\min}, x_i^{mid}] \cup \{x_i^{\max}\}$,
4. $\{x_i^{\min}\} \cup [x_i^{mid}, x_i^{\max}]$.

In the two latter cases, x_i^{mid} is a constant with $x_i^{\min} < x_i^{mid} < x_i^{\max}$.

Proof. If $a = 0$, then $f_i(x)$ is a convex function (if $b \geq 0$) or a concave function (if $b \leq 0$). For this reason, we only need to consider the case $a \neq 0$. We first seek the locally convex points x_0 since every globally convex point is also locally convex. Let $f_i^{(n)}$ denote the nth derivative of f_i. We have

$$\begin{aligned} f_i^{(1)}(x) &= 3ax^2 + 2bx + c, \\ f_i^{(2)}(x) &= 6ax + 2b, \\ f_i^{(3)}(x) &= 6a \neq 0, \\ f_i^{(n)}(x) &= 0 \text{ for all } n \geq 4. \end{aligned}$$

Similar to the proof of Lemma 5.5, using Taylor's Formula, we can easily prove that for any $x \in (x_i^{\min}, x_i^{\max})$, x is locally convex if $f_i^{(2)}(x) = 6ax + 2b > 0$. Note that the extreme points $x_i^{\min}$ and $x_i^{\max}$ are globally convex and thus locally convex which is implied by Corollary 5.22. Since $f_i^{(2)}(x) = 6ax + 2b$ is a monotonic function and has at most one root, depending on the value of a, b $x_i^{\min}$ and $x_i^{\max}$, the set of locally convex domain $\check{X}_i^l$ has one of the following four forms:

1. $\{x_i^{\min}, x_i^{\max}\}$,
2. $[x_i^{\min}, x_i^{\max}]$,
3. $[x_i^{\min}, -b/3a) \cup \{x_i^{\max}\}$,

4. $\{x_i^{\min}\} \cup (-b/3a, x_i^{\max}]$.

We now discuss the set $\check{X}_i^g$ with the four cases above.

1. It is clear that $\check{X}_i^g = \{x_i^{\min}, x_i^{\max}\}$ if $\check{X}_i^l = \{x_i^{\min}, x_i^{\max}\}$.

2. If $\check{X}_i^l = [x_i^{\min}, x_i^{\max}]$, then $f_i^{(2)}(x) \geq 0$ for all $x \in [x_i^{\min}, x_i^{\max}]$ which implies that $f_i(x)$ is a convex function with domain $[x_i^{\min}, x_i^{\max}]$. Since $f_i(x)$ is differentiable, the tangent plane $\{(x, y) \mid y = (3ax_0^2 + 2bx_0 + c)(x - x_0) + f_i(x_0)\}$ at point $(x_0, f_i(x_0))$ for every $x_0 \in F_i$ is below the graph of f_i over F_i. Hence $\check{X}_i^g = [x_i^{\min}, x_i^{\max}]$.

3. Examples of this case can be seen in Figure 5.14. Note that $\check{X}_i^g \subset [x_i^{\min}, -b/3a) \cup \{x_i^{\max}\}$ and $\{x_i^{\min}, x_i^{\max}\} \subset \check{X}_i^g$. Theorem 5.8 implies that $x_0 \in (x_i^{\min}, -b/3a)$ is globally convex if and only if the corresponding tangent plane

   $$T_d(x_0) = \{(x, y) \mid y = (3ax_0^2 + 2bx_0 + c)(x - x_0) + f_i(x_0)\}$$

   is valid. Note that $T_d(x_0)$ is below the graph of f_i in $[x_i^{\min}, -b/3a]$. With $\check{X}_i^g \subset [x_i^{\min}, -b/3a) \cup \{x_i^{\max}\}$, Lemma 5.20 implies that $x_0 \in (x_i^{\min}, -b/3a)$ is globally convex if and only if $T_d(x_0)$ is below the point $(x_i^{\max}, f_i(x_i^{\max}))$. Define

   $$g_{\max}(x) = (3ax^2 + 2bx + c)(x_i^{\max} - x) + f_i(x)$$

   such that point $(x_i^{\max}, g_{\max}(x)) \in T_d(x)$ for any $x \in [x_i^{\min}, -b/3a]$. The tangent plane $T_d(x_0)$ for $x_0 \in [x_i^{\min}, -b/3a]$ is below the point $(x_i^{\max}, f_i(x_i^{\max}))$ if and only if $g_{\max}(x_0) \leq f_i(x_i^{\max})$. Thus we only need to compare $g_{\max}(x_0)$ and $f_i(x_i^{\max})$. Consider the first derivative of $g_{\max}(x)$

   $$\begin{aligned} g_{\max}^{(1)}(x) &= (x_i^{\max} - x)(6ax + 2bx) - (3ax^2 + 2bx + c) + f_i^{(1)}(x) \\ &= (x_i^{\max} - x)(6ax + 2bx). \end{aligned}$$

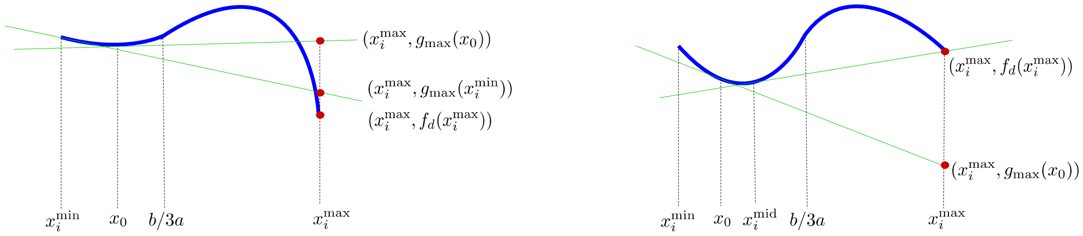
   Thus, $g_{\max}$ is strictly increasing on $[x_i^{\min}, -b/3a)$ since we have $g_{\max}^{(1)}(x) > 0$ for any $x \in [x_i^{\min}, -b/3a)$; similarly $g_{\max}$ is strictly decreasing on $(-b/3a, x_i^{\max})$ since $g_{\max}^{(1)}(x) < 0$ for any $x \in [x_i^{\min}, -b/3a)$. It is then clear that

   $$g_{\max}(-b/3a) > g_{\max}(x_i^{\max}) = f_i(x_i^{\max}).$$

   Now we compare $g_{\max}(x_i^{\min})$ and $f_i(x_i^{\max})$. If $g_{\max}(x_i^{\min}) > f_i(x_i^{\max})$, see an example in Figure 5.14a, we have $g_{\max}(x) > f_i(x_i^{\max})$ for all $x \in (x_i^{\min}, -b/3a)$. Hence no point in $(x_i^{\min}, -b/3a)$ is globally convex, which implies $\check{X}_i^g = \{x_i^{\min}, x_i^{\max}\}$.

   Otherwise we have $g_{\max}(x_i^{\min}) \leq f_i(x_i^{\max})$, see an example in Figure 5.14b. Consider the strictly increasing function $g_{\max}(x) - f_i(x_i^{\max})$ over $(x_i^{\min}, -b/3a)$, with $g_{\max}(x_i^{\min}) - f_i(x_i^{\max}) \leq 0$ and $g_{\max}(-b/3a) - f_i(x_i^{\max}) > 0$. This function has exactly one real root over $[x_i^{\min}, -b/3a)$, say x_i^{mid}. Then we have $\check{X}_i^g = [x_i^{\min}, x_i^{\text{mid}}] \cup \{x_i^{\max}\}$ if $x_i^{\min} < x_i^{\text{mid}}$ and $\check{X}_i^g = \{x_i^{\min}, x_i^{\max}\}$ if $x_i^{\min} = x_i^{\text{mid}}$.

4. Similar to case 3, we need only to know whether the polynomial function of x
$$\underbrace{(3ax^2 + 2bx + c)(x_i^{\min} - x) + f_i(x)}_{=:g_{\min}(x)} - f_i(x_i^{\min})$$
has a real root over $x \in (-b/3a, x_i^{\max})$. If the root exists, say x_i^{mid}, then we have $\check{X}_i^g = \{x_i^{\min}\} \cup [x_i^{\text{mid}}, x_i^{\max}]$; otherwise, we have $\check{X}_i^g = \{x_i^{\min}, x_i^{\max}\}$ as well. □



(a) Case 1

(b) Case 2

Figure 5.14: Examples for globally and locally convex domain points

Considering the four cases, the set of globally convex points $\check{X}_i^g \subset [x_i^{\min}, x_i^{\max}]$ has either two points, or is an interval plus a point, or an interval. Since the projection function g_d^{-1} is bijective, the set of globally convex points on F_i, denoted by $\check{X}^g \cap F_i = g_d^{-1}(\check{X}_i^g)$, also consists of either two extreme points, or is a line segment in $\mathbb{R}^2$ plus an extreme point, or a line segment in $\mathbb{R}^2$. Note that every extreme point of X is globally convex. We call an extreme point an *isolated* extreme point if it is not contained in a line segment that consists of globally convex boundary domain points only. We then get the following lemma easily.

Lemma 5.34
The set $\check{X}^g \cap \partial X$ of globally convex boundary domain points for the graph of $f(x, y)$ over the polytope $X \in \mathbb{R}^2$ is a union of m_1 line segments and m_2 isolated extreme points with $m_1, m_2 \in \mathbb{N}_0, m_1 \leq m$ and $m_2 \leq m$.

Let $L_1, L_2, \ldots, L_{m_1}$ be the m_1 line segments and $\mathbf{x}_1^e, \mathbf{x}_2^e, \ldots, \mathbf{x}_{m_2}^e$ be the m_2 isolated extreme points. With this notation we have
$$\check{X}^g \cap \partial X = L_1 \cup \cdots \cup L_{m_1} \cup \{\mathbf{x}_1^e, \ldots, \mathbf{x}_{m_2}^e\}.$$

Furthermore, let $\mathcal{S}_{L_i}$ be the graph of $f(x, y)$ on L_i with
$$\mathcal{S}_{L_i} = \{(x, y, z) \mid z = f(x, y), (x, y) \in L_i\}$$

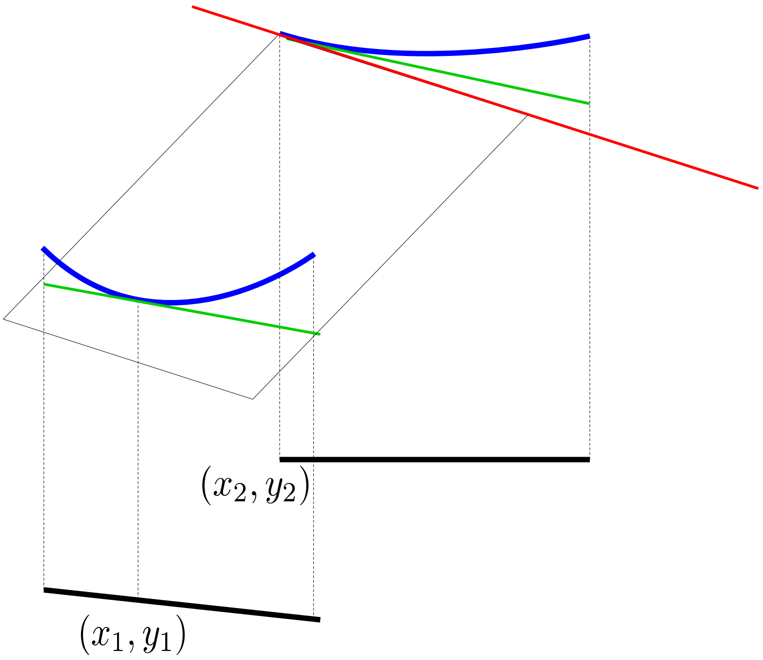


Figure 5.15: Hyperplane that intersects $\mathcal{S}_{L_i}$, $\mathcal{S}_{L_j}$ and below them

for every $i \in \{1, \dots, m\}$ and let

$$\mathcal{S}_{X^e} = \{(x, y, z) \mid z = f(x, y), (x, y) \in X^e\}.$$

In the following, for any $(x_0, y_0) \in L_i$, we show that there exists a hyperplane H through $(x_0, y_0, f(x_0, y_0))$ such that H is below $\mathcal{S}_{L_i}$ over $\check{X}^g \cap \partial X$. In Lemma 5.36 we have more details included. We show later in Theorem 5.37 that either H is a tight valid hyperplane or a tight valid hyperplane H^* can be found very easily which is parallel to H.

To find the hyperplane H with the properties described above, we first prove Lemma 5.35, which implies that for any L_j there exists a hyperplane H_i^j which is below $\mathcal{S}$ over $L_i \cup L_j$ with $i, j \in \{1, \dots, m_1\}, i \neq j$. Using this result, we show that a hyperplane H through $(x_0, y_0, f(x_0, y_0))$ exists such that H is below $\mathcal{S}_{L_j}$ for any $j \in \{1, \dots, m_1\}, i \neq j$. In addition, a hyperplane H can be found that it is below $(\mathbf{x}_k^e, f(\mathbf{x}_k^e))$ for any $k \in \{1, \dots, m_1\}$.

Lemma 5.35

For any L_i and L_j with $i, j \in \{1, \dots, m_1\}, i \neq j$ and for any $(x_0, y_0) \in L_i$, there exists a hyperplane H through $(x_0, y_0, f(x_0, y_0))$ with $H \cap \mathcal{S}_{L_j} \neq \emptyset$ and H is below $\mathcal{S}_{L_i}$ and $\mathcal{S}_{L_j}$.

Moreover, such a hyperplane H is unique for any $(x_0, y_0) \in L_i \setminus X^e$.

Proof. An example is shown in Figure 5.15. The two blue curves are $\mathcal{S}_{L_i}$ and $\mathcal{S}_{L_j}$. We need to find a hyperplane H through $(x_0, y_0, f(x_0, y_0))$ that intersects both $\mathcal{S}_{L_i}$ and $\mathcal{S}_{L_j}$ and at the same time H is below them.

For the special case $(x_0, y_0) \in L_j$, we can easily check that $H = T(x_0, y_0)$, i.e., the tangent plane at (x_0, y_0) fulfills all the conditions. In this case H is not unique.

Assume that $(x_0, y_0) \notin L_j$. We discuss the case $(x_0, y_0) \in L_i \setminus X^e$, e.g., $(x_0, y_0) = (x_1, y_1)$ in Figure 5.15. Corollary 5.13 implies that a hyperplane H through $(x_0, y_0, f(x_0, y_0))$ which is below $\mathcal{S}_{L_i}$ contains the subtangent plane

$$\text{Ł}(x_0, y_0) = T(x_0, y_0) \cap \{(x, y, z) \mid (x, y) \in \operatorname{aff}\{L_i\}\} \tag{5.18}$$

which is the lower left green line in Figure 5.15. Denote $P_0 = (x_0, y_0, f(x_0, y_0))$. For every point $P_j = (x_j, y_j, z_j) \in \mathcal{S}_{L_j}$, we define $H(\text{Ł}(x_0, y_0), P_j) = \operatorname{aff}\{\text{Ł}(x_0, y_0), \{P_j\}\}$ which is a hyperplane below P_j. Similar to the proof of Theorem 5.27, there exists a point $P^* \in \mathcal{S}_{L_j}$ such that $H^* = H(\text{Ł}(x_0, y_0), P^*)$ is below $\mathcal{S}_{L_j}$. Note that H^* is unique since it is associated to the objective value of an optimization problem introduced in Theorem 5.27 which always has an optimum.

Finally, we discuss the case $(x_0, y_0) \in L_i \cap X^e$, e.g., $(x_0, y_0) = (x_2, y_2)$ in Figure 5.15. Consider a line $\text{Ł}^l(x_0, y_0) \subset \{(x, y, z) \mid (x, y) \in \operatorname{aff}\{L_i\}\}$ through $(x_0, y_0, f(x_0, y_0))$ which is below $\text{Ł}(x_0, y_0)$ defined by (5.18) such that $(x, y) \in L_i$. In the example in Figure 5.15, $\text{Ł}^l(x_0, y_0)$ is the red line and $\text{Ł}(x_0, y_0)$ is the upper right green line. Every hyperplane H which contains $\text{Ł}^l(x_0, y_0)$ is through $(x_0, y_0, f(x_0, y_0))$ and below $\mathcal{S}_{L_i}$. Similar to the discussion above, there exists a point $P^* \in \mathcal{S}_{L_j}$ such that $H^* = \operatorname{aff}\{\text{Ł}^l(x_0, y_0), \{P_j\}\}$ is below $\mathcal{S}_{L_j}$. Note that for every fixed chosen $\text{Ł}^l(x_0, y_0)$ there exists a unique H^*. However, we have infinitely many $\text{Ł}^l(x_0, y_0)$ to choose. □

Now we discuss how to algorithmically find H which fulfills Lemma 5.35. Note that for $(x_0, y_0) \in L_i \cap X^e$ we may choose $\text{Ł}^l(x_0, y_0) = \text{Ł}(x_0, y_0)$ which can be computed easily. For any $(x_0, y_0) \in L_i$ we compute a hyperplane H that fulfills Lemma 5.35 and satisfies $H \supset \text{Ł}(x_0, y_0)$ which is a line defined in (5.18). This is equivalent to finding a point $(x^*, y^*) \in L_j$ such that $H = \operatorname{aff}\{\text{Ł}(x_0, y_0), \{((x^*, y^*), f(x^*, y^*))\}\}$ is below $\mathcal{S}_{L_j}$. Consider the two lines $\operatorname{aff}\{L_i\}$ and $\operatorname{aff}\{L_j\}$. They are either not parallel or parallel. Examples for both cases are in Figure 5.16.

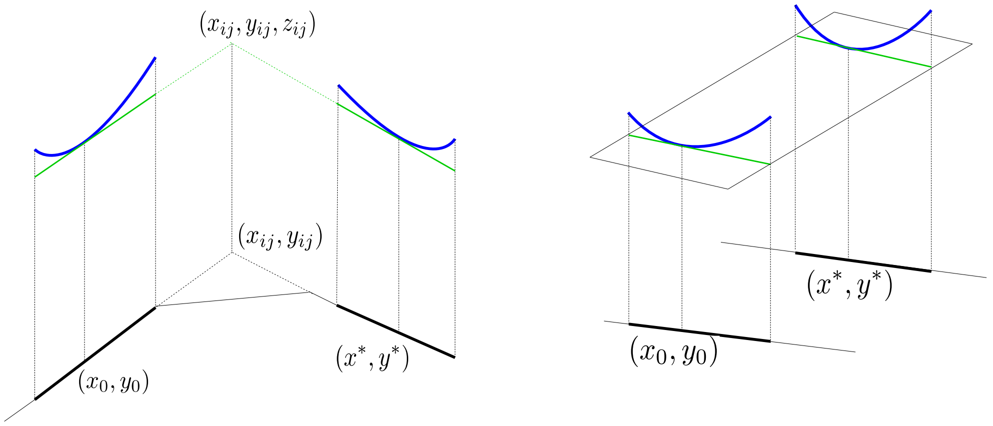
As mentioned before, for the case $(x_0, y_0) \in L_j$, we set $H = T(x_0, y_0)$ and we are done. Otherwise, let

$$L_i = \{(x, y) \mid y = a_i x + b_i, x \in [x_i^{\min}, x_i^{\max}]\}$$

and

$$L_j = \{(x, y) \mid y = a_j x + b_j, x \in [x_j^{\min}, x_j^{\max}]\}.$$

Define $f_{L_i}(x) = f(x, a_i x + b_i)$ for $x \in [x_i^{\min}, x_i^{\max}]$ and define $f_{L_j}(x) = f(x, a_j x + b_j)$ for $x \in [x_j^{\min}, x_j^{\max}]$. $f_{L_i}(x)$ and $f_{L_j}(x)$ are univariate functions with degree up to 3. The



(a) aff$\{L_i\}$ and aff$\{L_j\}$ are not parallel

(b) aff$\{L_i\}$ and aff$\{L_j\}$ are parallel

Figure 5.16: Two lines aff$\{L_i\}$ and aff$\{L_j\}$ can be parallel or not parallel

line $\text{Ł}_i(x_0, y_0) = T(x_0, y_0) \cap \{(x, y, z) \mid (x, y) \in \text{aff}\{L_i\}\}$ for $x_0 \in [x_i^{\min}, x_i^{\max}]$ with $y_0 = a_j x_0 + b_j$ can also be represented as

$$\text{Ł}_i(x_0, y_0) = \{(x, y, z) \mid x \in \mathbb{R}, y = a_i x + b_i, z = f'_{L_i}(x_0)(x - x_0) + f_{L_i}(x_0)\}. \tag{5.19}$$

Analogously, for every $x_1 \in [x_j^{\min}, x_j^{\max}]$ and $y_1 = a_j x_1 + b_j$, we get

$$\text{Ł}_j(x_1, y_1) = \{(x, y, z) \mid x \in \mathbb{R}, y = a_j x + b_j, z = f'_{L_j}(x_1)(x - x_1) + f_{L_j}(x_1)\}.$$

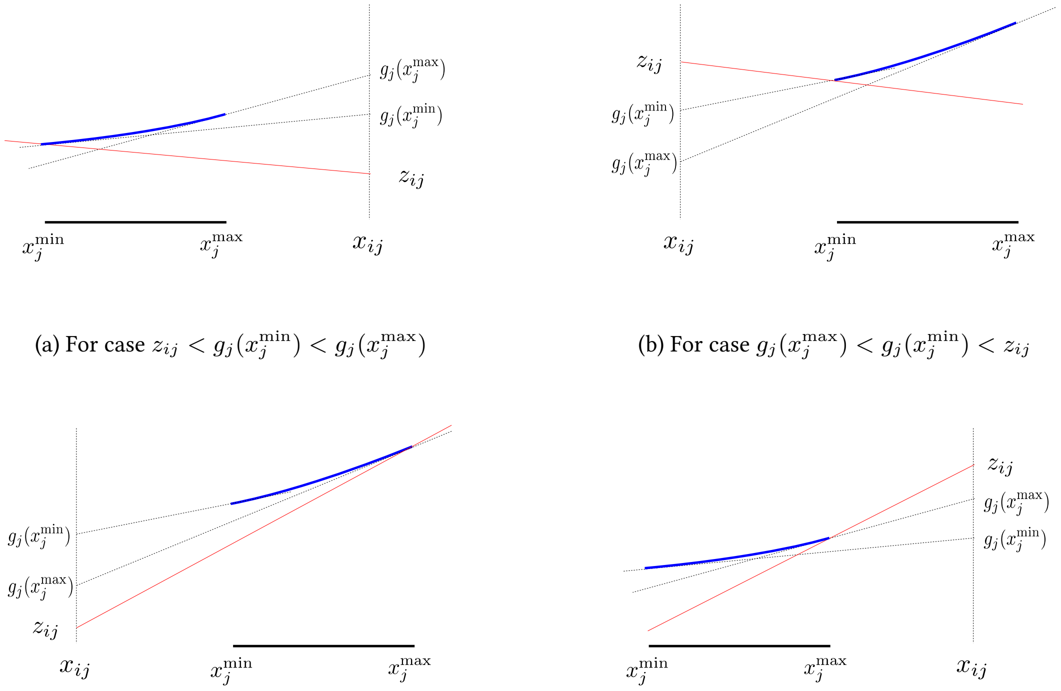
First we discuss the case that aff$\{L_i\}$ and aff$\{L_j\}$ are not parallel, see an example in Figure 5.16a. Since $a_i \neq a_j$, the intersection of aff$\{L_i\}$ and aff$\{L_j\}$ is (x_{ij}, y_{ij}) with

$$x_{ij} = \frac{b_j - b_i}{a_i - a_j} \text{ and } y_{ij} = \frac{b_j a_i - b_i a_j}{a_i - a_j}.$$

Consider the point $P_{ij} = (x_{ij}, y_{ij}, z_{ij})$ with $z_{ij} = f'_{L_i}(x_0)(x_{ij} - x_0) + f_{L_i}(x_0)$. We can check that $P_{ij} \in \text{Ł}_i(x_0, y_0)$ which implies that $P_{ij} \in H$ for every H that fulfills Lemma 5.35. Since H also intersects $\mathcal{S}_{L_j}$, finding H fulfilling Lemma 5.35 is equivalent to finding a point $P^* \in \mathcal{S}_{L_j}$ such that aff$\{\text{Ł}_i(x_0, y_0), \{P^*\}\}$ is below $\mathcal{S}_{L_j}$. Consider the function

$$g_j(x) = f'_{L_j}(x)(x_{ij} - x) + f_{L_j}(x)$$

for $x \in [x_j^{\min}, x_j^{\max}]$. Note that the point $(x_1, a_j x_1 + b_j, g_j(x_1))$ lies in line $\text{Ł}_j(x_1, y_1)$ for $x_1 \in [x_j^{\min}, x_j^{\max}]$. As we analyzed before by considering the sign of the first derivative, $g_j(x)$ is a strictly decreasing function if $x_{ij} < x_j^{\min}$ and is a strictly increasing function if



(a) For case $z_{ij} < g_j(x_j^{\min}) < g_j(x_j^{\max})$

(b) For case $g_j(x_j^{\max}) < g_j(x_j^{\min}) < z_{ij}$

(c) For case $z_{ij} < g_j(x_j^{\max}) < g_j(x_j^{\min})$

(d) For case $g_j(x_j^{\min}) < g_j(x_j^{\max}) < z_{ij}$

Figure 5.17: Example for the 4 cases in the proof of Lemma 5.35

$x_{ij} > x_j^{\max}$. There are no other cases since $x_{ij} \notin [x_j^{\min}, x_j^{\max}]$. In both cases we have $g_j(x_j^{\min}) \neq g_j(x_j^{\max})$. We compare z_{ij}, $g_j(x_j^{\min})$ and $g_j(x_j^{\max})$. If z_{ij} is between $g_j(x_j^{\min})$ and $g_j(x_j^{\max})$, i.e., $g_j(x_j^{\min}) \leq z_{ij} \leq g_j(x_j^{\max})$ or $g_j(x_j^{\max}) \leq z_{ij} \leq g_j(x_j^{\min})$, then the increasing or decreasing function $g_j(x) - z_{ij}$ has a unique root $x^* \in [x_j^{\min}, x_j^{\max}]$ with $g_j(x^*) - z_{ij} = 0$. Note that since $g_j(x)$ is a polynomial function of degree up to 3, x^* can be computed very easily. It implies that the line $\text{Ł}_j(x^*, y^*)$ with $y^* = a_j x^* + b_j$ contains also P_{ij} and

$$H = \operatorname{aff}\{\text{Ł}_i(x_0, y_0), \text{Ł}_j(x^*, y^*)\}$$

is a hyperplane which fulfills Lemma 5.35.

Otherwise, if z_{ij} is not between $g_j(x_j^{\min})$ and $g_j(x_j^{\max})$, we have the following four cases

1. $z_{ij} < g_j(x_j^{\min}) < g_j(x_j^{\max})$,

2. $g_j(x_j^{\max}) < g_j(x_j^{\min}) < z_{ij}$,

3. $z_{ij} < g_j(x_j^{\max}) < g_j(x_j^{\min})$,
4. $g_j(x_j^{\min}) < g_j(x_j^{\max}) < z_{ij}$.

Examples for the four cases are shown in Figure 5.17. In the first two cases we set

$$P^* = (x_j^{\min}, a_j x_j^{\min} + b_j, f_{L_j}(x_j^{\min})).$$

In the last two cases we set

$$P^* = (x_j^{\max}, a_j x_j^{\max} + b_j, f_{L_j}(x_j^{\max})).$$

For all four cases, the line $\operatorname{aff}\{P^*, P^{ij}\}$, shown as the red line in the corresponding subgraphs, is below $\mathcal{S}_{L_j}$ (the corresponding blue curve). We then have $H = \operatorname{aff}\{\text{Ł}_i(x_0, y_0), \{P^*\}\} \supset \operatorname{aff}\{P^*, P^{ij}\}$ which fulfills Lemma 5.35 and is the hyperplane we are looking for.

Now it only remains to discuss the case that $\operatorname{aff}\{L_i\}$ and $\operatorname{aff}\{L_j\}$ are parallel, see an example in Figure 5.16b. Note that $a_i = a_j$ and for any $x_1 \in [x_j^{\min}, x_j^{\max}]$ with $y_1 = a_j x_1 + b_j$, $\text{Ł}_j(x_1, y_1)$ is contained in the hyperplane $\{(x, y, z) \mid y = a_j x + b_j\}$. The other hyperplane $\{(x, y, z) \mid y = a_i x + b_i\}$ which contains $\text{Ł}_i(x_0, y_0)$ is parallel to the hyperplane $\{(x, y, z) \mid y = a_j x + b_j\}$. It implies that $\operatorname{aff}\{\text{Ł}_i(x_0, y_0), \text{Ł}_j(x_1, y_1)\}$ is a hyperplane if and only if $\text{Ł}_i(x_0, y_0)$ and $\text{Ł}_j(x_1, y_1)$ are parallel. This is equivalent to $f'_{L_j}(x_1) = f'_{L_i}(x_0)$. Note that $f_{L_j}(x)$ is a convex function for $x \in [x_j^{\min}, x_j^{\max}]$ and is a polynomial function with degree up to 3. Then $f'_{L_j}(x)$ is an increasing function with $f'_{L_j}(x_j^{\min}) < f'_{L_j}(x_j^{\max})$. Compare $f'_{L_i}(x_0)$, $f'_{L_j}(x_j^{\min})$ and $f'_{L_j}(x_j^{\max})$. If $f'_{L_j}(x_j^{\min}) \leq f'_{L_i}(x_0) \leq f'_{L_j}(x_j^{\max})$, then the function $f'_{L_j}(x) - f'_{L_i}(x_0)$ has a unique root x^* for $x \in [x_j^{\min}, x_j^{\max}]$. The function $f'_{L_j}(x)$ is a polynomial function with degree up to 2, for which we can easily compute the root x^*. In this case we set $P^* = (x^*, a_j x^* + b_j, f_{L_j}(x^*))$. See the example in Figure 5.16b again. Otherwise if $f'_{L_i}(x_0) < f'_{L_j}(x_j^{\min})$ we set $P^* = (x_j^{\min}, a_j x_j^{\min} + b_j, f_{L_j}(x_j^{\min}))$ and if $f'_{L_i}(x_0) > f'_{L_j}(x_j^{\max})$ we set $P^* = (x_j^{\max}, a_j x_j^{\max} + b_j, f_{L_j}(x_j^{\max}))$. Examples for both cases are shown in Figure 5.18. We can easily prove that the line $\operatorname{aff}\{\text{Ł}_i(x_0, y_0), \{P^*\}\} \cap \{(x, y, z) \mid (x, y) \in \operatorname{aff}\{L_j\}\}$, shown as the red line in the corresponding subgraphs, is below $\mathcal{S}_{L_j}$ (the corresponding blue curve). Thus $H = \operatorname{aff}\{\text{Ł}_i(x_0, y_0), \{P^*\}\}$ fulfills the requirements of Lemma 5.35 and is the hyperplane we are looking for.

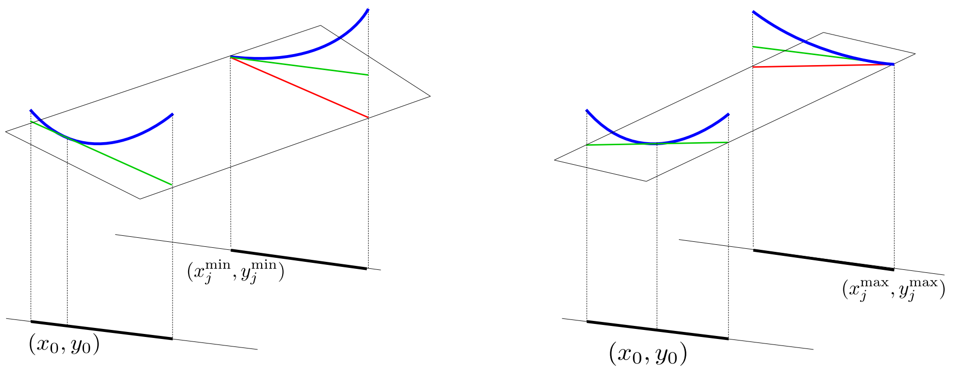
Consequently, for all cases above, we can always find a unique point $P^* \in \mathcal{S}_{L_j}$ such that $H = \operatorname{aff}\{\text{Ł}_i(x_0, y_0), \{P^*\}\}$ fulfills Lemma 5.35. Note that $P_i^{\min} = (x_i^{\min}, a_i x_i^{\min} + b_i, f_{L_i}(x_i^{\min}))$ and $P_i^{\max} = (x_i^{\max}, a_i x_i^{\max} + b_i, f_{L_i}(x_i^{\max}))$ are two different points on $\mathcal{S}_{L_i}$. Since $P_i^{\min}$, $P_i^{\max}$ and P^* do not lie on the same line, we set $H = \operatorname{aff}\{P_i^{\min}, P_i^{\max}, P^*\}$. We can compute H by solving a system of linear equations. The algorithm is summarized in Algorithm 5.1.

Define $\partial\mathcal{S} = \{(x, y, z) \mid (x, y) \in \partial X, z = f(x, y)\}$ and for each $i \in \{1, \dots, m\}$ and the corresponding facet F_i we further define $\mathcal{S}_{F_i} = \{(x, y, z) \mid (x, y) \in F_i, z = f(x, y)\}$.

Lemma 5.36
For any $L_i \subset \check{X}^g \cap \partial X$ with $i \in \{1, \dots, m_1\}$ and for any $(x_0, y_0) \in L_i \subset F_i$, there exists a hyperplane H

Algorithm 5.1: Algorithm that computes a hyperplane that intersects $\mathcal{S}_{L_i}, \mathcal{S}_{L_j}$ and is below them

```
Input: Line segments $L_i$ and $L_j$ with $i, j \in \{1, \dots, m_1\}$, a point $\mathbf{x}^0 = (x_0, y_0) \in L_i$
Output: The unique hyperplane $H(\mathbf{x}^0, L_i, L_j)$ with $H \supset \L_i(x_0, y_0)$, $H \cap \mathcal{S}_{L_j} \neq \emptyset$ and $H$
        is below $\mathcal{S}_{L_i}$ and $\mathcal{S}_{L_j}$
if aff$\{L_i\}$ and aff$\{L_j\}$ are not parallel then
    if $\mathbf{x}^0 \in L_j$ then
        return $T(\mathbf{x}_0)$
    end
    Compute the intersection point $(x_{ij}, y_{ij})$, value $g_j(x_j^{\min})$ and $g_j(x_j^{\max})$;
    Compute $z_{ij} = f'_{L_i}(x_0)(x_{ij} - x_0) + f_{L_i}(x_0)$ and set $P_{ij} = (x_{ij}, y_{ij}, z_{ij})$;
    if $g_j(x_j^{\min}) \leq z_{ij} \leq g_j(x_j^{\max})$ or $g_j(x_j^{\max}) \leq z_{ij} \leq g_j(x_j^{\min})$ then
        Compute the unique root $x^*$ of $g_j(x) - z_{ij} = 0$ for $x \in [x_j^{\min}, x_j^{\max}]$;
        Set $P^* = (x^*, a_j x^* + b_j, f_{L_j}(x^*))$
    end
    if $z_{ij} < g_j(x_j^{\min}) < g_j(x_j^{\max})$ or $g_j(x_j^{\max}) < g_j(x_j^{\min}) < z_{ij}$ then
        Set $P^* = (x_j^{\min}, a_j x_j^{\min} + b_j, f_{L_j}(x_j^{\min}))$
    end
    if $z_{ij} < g_j(x_j^{\max}) < g_j(x_j^{\min})$ or $g_j(x_j^{\min}) < g_j(x_j^{\max}) < z_{ij}$ then
        Set $P^* = (x_j^{\max}, a_j x_j^{\max} + b_j, f_{L_j}(x_j^{\max}))$
    end
end
else
    Compute $f'_{L_j}(x_j^{\min})$, $f'_{L_j}(x_j^{\max})$ and $f'_{L_i}(x_0)$;
    if $f'_{L_j}(x_j^{\min}) \leq f'_{L_i}(x_0) \leq f'_{L_j}(x_j^{\max})$ then
        Compute the unique root $x^*$ of $f'_{L_j}(x) - f'_{L_i}(x_0)$ for $x \in [x_j^{\min}, x_j^{\max}]$;
        Set $P^* = (x^*, a_j x^* + b_j, f_{L_j}(x^*))$.
    end
    if $f'_{L_i}(x_0) < f'_{L_j}(x_j^{\min})$ then
        Set $P^* = (x_j^{\min}, a_j x_j^{\min} + b_j, f_{L_j}(x_j^{\min}))$
    end
    if $f'_{L_j}(x_j^{\max}) < f'_{L_i}(x_0)$ then
        Set $P^* = (x_j^{\max}, a_j x_j^{\max} + b_j, f_{L_j}(x_j^{\max}))$
    end
end
Compute $\L_i(x_0, y_0)$
Compute $H((x_0, y_0), L_i, L_j) = \text{aff}\{\L_i(x_0, y_0), \{P^*\}\}$ and return $H((x_0, y_0), L_i, L_j)$
```



(a) For the case $f'_{L_i}(x_0) < f'_{L_j}(x_j^{\min})$

(b) For the case $f'_{L_i}(x_0) > f'_{L_j}(x_j^{\max})$

Figure 5.18: Two cases by $f'_{L_i}(x_0) \notin [f'_{L_j}(x_j^{\min}), f'_{L_j}(x_j^{\max})]$ in Algorithm 5.1

1. *either with* $H = T(x_0, y_0)$
2. *or with* $(H \cap \partial \mathcal{S}) \setminus \mathcal{S}_{F_i} \neq \emptyset$

such that $H \supset \text{Ł}_i(x_0, y_0)$*. In addition,* H *is below* $\mathcal{S}$ *over* $(x, y) \in \check{X}^g \cap \partial X$*.*

Proof. We develop an algorithm to find the hyperplane H.

Denote $\operatorname{aff}\{F_i\} = \{(x, y) \mid a_i x + b_i x = c_i\}$ with three constants $a_i, b_i, c_i \in \mathbb{R}$. Since F_i is a facet of X, we have $X \subset \{(x, y) \mid a_i x + b_i x \geq c_i\}$ or $X \subset \{(x, y) \mid a_i x + b_i x \leq c_i\}$. Lemma 5.18 implies that for any two nonvertical H_1, H_2 with $H_1 \supset \text{Ł}_i(x_0, y_0)$ and $H_2 \supset \text{Ł}_i(x_0, y_0)$, either H_1 is below H_2 over X or H_2 is below H_1 over X. Recall that $\check{X}^g \cap \partial X = L_1 \cup \cdots \cup L_{m_1} \cup \{\mathbf{x}_1^e, \ldots, \mathbf{x}_{m_2}^e\}$. For every L_j with $j \neq i$ and $j \in \{1, \ldots, m_1\}$, compute $H_j^L = H((x_0, y_0), L_i, L_j)$ as output of Algorithm 5.1. For every $\mathbf{x}_j^e$ with $\mathbf{x}_j^e \notin F_i$ and $j \in \{1, \ldots, m_2\}$, compute $H_j^e = \operatorname{aff}\{\text{Ł}_i(x_0, y_0), \{(\mathbf{x}_j^e, f(\mathbf{x}_j^e))\}\}$ which is a nonvertical hyperplane since $(\mathbf{x}_j^e, f(\mathbf{x}_j^e)) \notin \text{Ł}_i(x_0, y_0)$ and $\mathbf{x}_j^e \notin F_i$. Consider the set

$$\mathcal{H}(x_0, y_0) = \{H_j^L \mid j \neq i, j \in \{1, \ldots, m_1\}\} \cup \{H_j^e \mid \mathbf{x}_j^e \notin F_i, j \in \{1, \ldots, m_2\}\}$$

of finitely many hyperplanes that all contain $\text{Ł}_i(x_0, y_0)$. There exists $H^* \in \mathcal{H}(x_0, y_0)$ such that H^* is below H over X for every $H \in \mathcal{H}(x_0, y_0)$. The hyperplane H^* is below $\mathcal{S}_{F_i}$ since $\text{Ł}_i(x_0, y_0) \subset H^*$ which is below $\mathcal{S}_{F_i}$. The hyperplane H^* is below every $\mathcal{S}_{F_j}$ with $j \neq i$ since H^* is below H_j^L over $F_j \subset X$ and H_j^L is below $\mathcal{S}_{F_j}$. Similarly, H^* is below every $(\mathbf{x}_j^e, f(\mathbf{x}_j^e))$ since H^* is below H_j^e over $X \ni \mathbf{x}_j^e$. Thus H^* is below $\mathcal{S}$ over $(x, y) \in \check{X}^g \cap \partial X$.

If $H^* = H_k^L$ for some $k \in \{1, \dots, m_1\}$, Algorithm 5.1 implies that either $H^* = T(x_0, y_0)$ or there exists a point $\mathbf{x}_k \in L_k$ with $\mathbf{x}_k \notin F_i$ and $(\mathbf{x}_k, f(\mathbf{x}_k)) \in H^*$. Otherwise if $H^* = H_k^e$, there exists a point $\mathbf{x}_k \in X^e$ with $\mathbf{x}_k \notin F_i$ and $(\mathbf{x}_k, f(\mathbf{x}_k)) \in H^*$.

In all cases either $H = T(x_0, y_0)$ or $(\mathbf{x}_k, f(\mathbf{x}_k)) \in (H \cap \partial \mathcal{S}) \setminus \mathcal{S}_{F_i} \neq \emptyset$. □

Theorem 5.37
For any $L_i \subset \check{X}^g \cap \partial X$ with $i \in \{1, \dots, m_1\}$ and for any $(x_0, y_0) \in L_i \subset F_i$, a hyperplane H^* which fulfills Lemma 5.36 is either a tight valid hyperplane or there exists a tight valid hyperplane H^{**} which is parallel to H^*. Furthermore, H^* is always a tight valid hyperplane if the Hessian matrix is negative semidefinite, i.e., it satisfies

$$H(f)(x,y) = \begin{pmatrix} \frac{\partial^2}{\partial x^2} f & \frac{\partial^2}{\partial x \partial y} f \\ \frac{\partial^2}{\partial x \partial y} f & \frac{\partial^2}{\partial y^2} f \end{pmatrix} \preceq 0 \tag{5.20}$$

for all $(x, y) \in \operatorname{int} X$.

Proof. Denote $H^* = \{(x, y, z) \mid z = a_i x + b_i y + c_i\}$ with $a_i, b_i, c_i \in \mathbb{R}$. Consider the optimization problem

$$\min_{(x,y) \in X} f(x,y) - (a_i x + b_i y + c_i) \tag{OP$_{H^*}^{\min}$}$$

that has a minimum z^*, since X is a compact set and $f(x, y) - (a_i x + b_i y + c_i)$ is a continuous function. Note that $z^* \leq 0$ since $\mathcal{S} \cap H^* \neq \emptyset$. The hyperplane H^* is valid if and only if $z^* = 0$. The maximally valid subtangent plane $T_H^{\max}(x_0, y_0)$ is $Ł_i(x_0, y_0)$ and there exists another point $(x_1, y_1, z_1) \in \mathcal{S} \cap H^*$ with $(x_1, y_1, z_1) \notin \operatorname{aff}\{Ł_i(x_0, y_0)\}$. Recalling the definition of tight valid hyperplanes in Section 5.3, we have

$$\operatorname{aff}\{Ł_i(x_0, y_0), \{(x_1, y_1, z_1)\}\} \subset \operatorname{aff}\left\{T_{H^*}^{\max}(\mathbf{x}^0) : \text{ for all } \mathbf{x}^0 \in X^{H^*}\right\} =: S^{H^*}$$

which implies

$$2 = \dim \operatorname{aff}\{Ł_i(x_0, y_0), \{(x_1, y_1, z_1)\}\} \leq \dim S^{H^*} \leq 2.$$

Hence S^{H^*} is a hyperplane which implies H^* is a tight valid hyperplane. Otherwise we have $z^* < 0$ with solution (x^*, y^*). Note that (x^*, y^*) must be an interior point of X since H^* is below $\mathcal{S}$ over $\check{X}^g \cap \partial X$. We have then $T(x^*, y^*) = a_i x + b_i y + c_i + z^*$ which is a tight valid hyperplane and is parallel to H^*.

Every globally convex interior point is also a locally convex interior point. According to [Edw94] and Lemma 5.5, (x_0, y_0) satisfies $H(f)(x_0, y_0) \succeq 0$. If (5.20) is satisfied for all $(x, y) \in \operatorname{int} X$ then we have $\check{X}^g \cap \operatorname{int} X = \emptyset$ which implies that H^* is below $\mathcal{S}$ over $\check{X}^g$. Lemma 5.20 implies that H^* is valid. As we discussed above, H^* is tight if it is valid. □

The hyperplanes fulfilling Lemma 5.36 are *potentially* tight valid hyperplanes since we only need to check if the corresponding optimization problem ($\mathrm{OP}_{H^*}^{\min}$) has the minimum $z^* = 0$. Until now we have only considered *potentially* tight valid hyperplanes that contain $Ł_i(x_0, y_0)$ for an (x_0, y_0) in an L_i. In order to show that there exists other *potentially* tight valid hyperplanes H which are below $f(x, y)$ over $\check{X}^g \cap \partial X$ we give the following definition.

Definition 5.38 (Potentially tight valid hyperplanes of type A and type B)
A hyperplane which fulfills Lemma 5.36 is a *potentially tight* valid hyperplanes of *type* A. A hyperplane H is a *potentially tight* valid hyperplane of *type* B if

- H is below $f(x, y)$ over $\check{X}^g \cap \partial X$,
- it satisfies $\pi_{\mathbf{x}}(H \cap \partial \mathcal{S}) \subset X^e$ and $|H \cap \partial \mathcal{S}| \geq 3$ and
- there does not exist L_i with $\mathbf{x}^e \in L_i$ and $\mathbf{x}^e \in \pi_{\mathbf{x}}(H \cap \partial \mathcal{S})$ such that $H \supset Ł_i(\mathbf{x}^e)$.

Due to the last condition in the definition of potentially tight valid hyperplanes of type B, the set of potentially tight valid hyperplanes of type A and the set of type B are disjoint.

Corollary 5.39
Let H^* be a potentially tight valid hyperplane of type B. Then H^* is either a tight valid hyperplane or there exists a tight valid hyperplane H^{**} which is parallel to H^*. Furthermore, H^* is always a tight valid hyperplane if every $(x, y) \in \operatorname{int} X$ satisfies (5.20).

Proof. The proof is the same as the proof of Theorem 5.37. □

Now we discuss how to compute potentially tight valid hyperplanes H^{**} of type B algorithmically. Note that every such H satisfies $\pi_{\mathbf{x}}(H \cap \partial \mathcal{S}) \subset X^e$ and $|H \cap \partial \mathcal{S}| \geq 3$. As every three points $\mathbf{x}^i, \mathbf{x}^j, \mathbf{x}^k \in X^e$ with $1 \leq i < j < k \leq m$ do not lie on a same line, it implies that $H^{ijk} = \operatorname{aff}\{(\mathbf{x}^i, f(\mathbf{x}^i), (\mathbf{x}^j, f(\mathbf{x}^j), (\mathbf{x}^k, f(\mathbf{x}^k)\}$ is a hyperplane. There are $\binom{m}{3}$ such hyperplanes H^{ijk}. We can easily prove that a given H^{ijk} is a potentially tight valid hyperplane H^{**} of type B if and only if

- for every $\mathbf{x}^l \in X^e$, H^{ijk} is below $(\mathbf{x}^l, f(\mathbf{x}^l))$;
- for every L_i, H^{ijk} is below $\mathcal{S}_{L_i}$, for this we need only to compare the curve $\mathcal{S}_{L_i}$ defined by a polynomial of degree up to 3 and the line segment $H^{ijk} \cap \{(\mathbf{x}, z) \mid \mathbf{x} \in L_i\}$;
- for any L_k containing $\mathbf{x}^s$, $s \in \{i, j, k\}$, check if it fulfills $Ł_i(\mathbf{x}^l) \not\subset H^{ijk}$.

All the three conditions above can be checked easily. Thus we design Algorithm 5.3 to compute potentially tight valid hyperplanes of type B.

The proof of Lemma 5.36 describes an algorithm to compute the unique potentially tight valid hyperplane H^* of type A that contains $Ł_i(\mathbf{x}_0)$, denoted by $H^*(\mathbf{x}_0, L_i)$. Note that we cannot omit L_i in the notation since there may exist $\mathbf{x}_k \in X^e$ with $\mathbf{x}_k \in L_i$, $\mathbf{x}_k \in L_j$, $i \neq j$

and $H^*(\mathbf{x}_k, L_i) \neq H^*(\mathbf{x}_k, L_j)$. For any $\mathbf{x}_1 \in L_j$ with $\mathbf{x}_1 \neq \mathbf{x}_0$, $H^*(\mathbf{x}_0, L_i) = H^*(\mathbf{x}_1, L_j)$ if and only if $H^*(\mathbf{x}_0, L_i) \supset Ł_j(\mathbf{x}_1)$. On the other hand, for every three points $\mathbf{x}^i, \mathbf{x}^j, \mathbf{x}^k \in X^e$, we use the algorithm above to check if $H^{ijk} = \operatorname{aff}\{(\mathbf{x}^i, f(\mathbf{x}^i), (\mathbf{x}^j, f(\mathbf{x}^j), (\mathbf{x}^k, f(\mathbf{x}^k)\}$ is a potentially tight valid hyperplane H^{**} of type B. If yes, denote it by $H^{**}(i,j,k)$. Let $\mathbf{x}^{i'}, \mathbf{x}^{j'}, \mathbf{x}^{k'} \in X^e$ be three points such that $H^{i'j'k'}$ is a potentially tight valid hyperplane of type B with $\{\mathbf{x}^{i'}, \mathbf{x}^{j'}, \mathbf{x}^{k'}\} \neq \{\mathbf{x}^i, \mathbf{x}^j, \mathbf{x}^k\}$. Then $H^{**}(i,j,k) = H^{**}(i',j',k')$ if and only if all the points $\{(\mathbf{x}^l, f(\mathbf{x}^l)) \mid l \in \{i,j,k,i',j',k'\}\}$ are on a same hyperplane.

Definition 5.40 (Tight valid hyperplanes of type A and type B)
A tight valid hyperplane H^* is a tight valid hyperplane of type A if H^* is also a potentially tight valid hyperplane of type A or H^* is parallel to a potentially tight valid hyperplanes of type A. Similarly, a tight valid hyperplane H^{**} is a tight valid hyperplane of type B if H^{**} is also a potentially tight valid hyperplane of type B or H^{**} is parallel to a potentially tight valid hyperplane of type B.

For any $i \in \{1, \ldots, m_1\}$, let $X_i \subset L_i$ be a set of finitely many points. Algorithm 5.2 computes a set of tight valid hyperplanes of type A. Let $N_2 \in \mathbb{N}$ be the upper bound of the number of tight valid hyperplanes of type B we want to have. Algorithm 5.3 computes a set of tight valid hyperplanes of type B.

5.5 Computational results

Recall the complete MINLP model (2.26) introduced in Section 2.1. All nonlinearities and integrality conditions can be handled by the solver SCIP directly. Note that in that model, we just consider pumps with fixed speed because of the two real-world instances introduced in Section 3.3.

In many water supply networks, there are variable speed pumps. For them the characteristic diagrams often involve the relative speed ω. In [Hae08; Kol11], the pressure increase is approximated by

$$\Delta h_{pt} = \omega_{pt}^2 \alpha_{0p} - \omega_{pt}\alpha_{1p}Q_{pt} - \alpha_{2p}Q_{pt}^2, \tag{5.21}$$

where α_{0p}, α_{1p} and α_{2p} are constants derived from the characteristic curve for pump p.

Figure 5.19 shows the characteristic curves for pump p with variable speed, in cases of $\omega_1 = 1$, $\omega_1 = 0.8$ and $\omega_1 = 0.6$.

Similar to (2.23), the power consumption of pump p can be approximated as

$$C_{pt} = \frac{\kappa_t \rho g \Delta h_{pt} Q_{pt}}{\eta_{pt}} = \frac{\kappa_t \rho g \left(\omega_{pt}^2 \alpha_{0p} Q_{pt} - \omega_{pt}\alpha_{1p}Q_{pt}^2 - \alpha_{2p}Q_{pt}^3\right)}{\eta_{pt}(Q_{pt}, \omega_{pt})} =: g(Q_{pt}, \omega_{pt}) \tag{5.22}$$

Note that the efficiency η_{pt} also depends on Q_{pt} and ω_{pt} and there exists a function to present it, hence there exists a function $g(Q_{pt}, \omega_{pt})$ to approximate C_{pt}.

Algorithm 5.2: Algorithm that computes a set of tight valid hyperplanes of type A

```
Input: A polynomial function in form (5.17), polytope $X \subset \mathbb{R}^2$ as the domain set, the
       corresponding $L_1, L_2, \ldots, L_{m_1}$ and $\{\mathbf{x}_1^e, \ldots, \mathbf{x}_{m_2}^e\}$, point sets $X_1, X_2, \ldots, X_{m_1}$
Output: Set $\mathcal{H}^A$ of tight valid hyperplanes for $\mathcal{S}$
Initialize $\mathcal{H}^A = \mathcal{H}^* = \emptyset$
for every $L_i \in \{L_1, L_2, \ldots, L_{m_1}\}$ do
    Compute $H^* := H(\mathbf{x}^0, L_i, L_{i'})$ for an $L_{i'} \neq L_i$ using Algorithm 5.1;
    for every $\mathbf{x}^0 \in L_i$ do
        for every $L_j \in \{L_1, \ldots, L_{m_1}\}$ with $L_j \neq L_i$ and $L_j \neq L_{i'}$ do
            Compute $H(\mathbf{x}^0, L_i, L_j)$ using Algorithm 5.1;
            if $H(\mathbf{x}^0, L_i, L_j)$ is below $H^*$ over $X$ then
                Set $H^* = H(\mathbf{x}^0, L_i, L_j)$
            end
        end
        Compute $Ł_i(\mathbf{x}^0)$ defined by (5.19);
        for every $\mathbf{x}_i^e \in \{\mathbf{x}_1^e, \ldots, \mathbf{x}_{m_2}^e\}$ with $\mathbf{x}_i^e \notin F_i$ do
            Compute $H' = \operatorname{aff}\{Ł_i(\mathbf{x}^0), \{(\mathbf{x}_i^e, f(\mathbf{x}_i^e))\}\}$;
            if $H'$ is below $H^*$ over $X$ then
                Set $H^* = H'$
            end
        end
        Set $\mathcal{H}^* = \mathcal{H}^* \cup \{H^*\}$
    end
end
if every $\mathbf{x}$ int $X$ satisfies (5.20) then
    Set $\mathcal{H}^A = \mathcal{H}^*$
end
else
    for every $H^* = \{(x, y, z) \mid z = a_i x + b_i y + c\} \in \mathcal{H}^*$ do
        Solve ($\mathrm{OP}_{H^*}^{\min}$) and get optimum $z^*$;
        if $z^* = 0$ then
            Set $\mathcal{H}^A = \mathcal{H}^A \cup \{H^*\}$
        end
        else
            Set $\mathcal{H}^A = \mathcal{H}^A \cup \{\{(x, y, z) \mid z = a_i x + b_i y + c + z^*\}\}$
        end
    end
end
Return $\mathcal{H}^A$
```

Algorithm 5.3: Algorithm that computes a set of tight valid hyperplanes of type B

```
   Input: A polynomial function in form (5.17), domain set $X \subset \mathbb{R}^2$ and $N_2 \in \mathbb{N}$
   Output: A set $\mathcal{H}$ of up to $N_2$ tight (downward closed) valid hyperplanes for $\mathcal{S}$
 1 Initialize $\mathcal{H}^B = \mathcal{H}^{**} = \emptyset$
 2 for every $\mathbf{x}^i, \mathbf{x}^j, \mathbf{x}^k \in X^e$ with $1 \le i < j < k \le m$ and $|\mathcal{H}^{**}| < N_2$ do
 3     Compute $H^{ijk} = \operatorname{aff}\{(\mathbf{x}^i, f(\mathbf{x}^i), (\mathbf{x}^j, f(\mathbf{x}^j), (\mathbf{x}^k, f(\mathbf{x}^k)\}$ and set $\mathcal{H}' = \{H^{ijk}\}$;
 4     for every $\mathbf{x}^l \in X^e$ do
 5         if "$H^{ijk}$ is below $(\mathbf{x}^l, f(\mathbf{x}^l))$" is false then
 6             Set $\mathcal{H}' = \emptyset$ and Goto line 23
 7         end
 8     end
 9     for every $L_i \in \{L_1, L_2, \dots, L_{m_1}\}$ do
10         if "$H^{ijk}$ is below $\mathcal{S}_{L_i}$" is false then
11             Set $\mathcal{H}' = \emptyset$ and Goto line 23
12         end
13         if $(H^{ijk} \cap \mathcal{S}_{L_i}) \subset \mathcal{S}_{X^e}$ then
14             Compute the potential unique point $(\mathbf{x}^*, f(\mathbf{x}^*)) \in H^{ijk} \cap \mathcal{S}_{L_i}$;
15             if $(\mathbf{x}^*, f(\mathbf{x}^*))$ exists and $H^{ijk} \supset \text{Ł}_i(\mathbf{x}^*)$ then
16                 Set $\mathcal{H}' = \emptyset$ and Goto line 23
17             end
18         end
19         else
20             Set $\mathcal{H}' = \emptyset$ and Goto line 23
21         end
22     end
23     Set $\mathcal{H}^{**} = \mathcal{H}^{**} \cup \mathcal{H}'$
24 end
25 if every $\mathbf{x} \in \operatorname{int} X$ satisfies (5.20) then
26     Set $\mathcal{H}^B = \mathcal{H}^{**}$
27 end
28 else
29     for every $H^* = \{(x, y, z) \mid z = a_i x + b_i y + c\} \in \mathcal{H}^{**}$ do
30         Solve ($\mathrm{OP}_{H^*}^{\min}$) and get optimum $z^*$;
31         if $z^* = 0$ then
32             Set $\mathcal{H}^B = \mathcal{H}^B \cup \{H^*\}$
33         end
34         else
35             Set $\mathcal{H}^B = \mathcal{H}^B \cup \{\{(x, y, z) \mid z = a_i x + b_i y + c + z^*\}\}$
36         end
37     end
38 end
39 Return $\mathcal{H}^B$
```

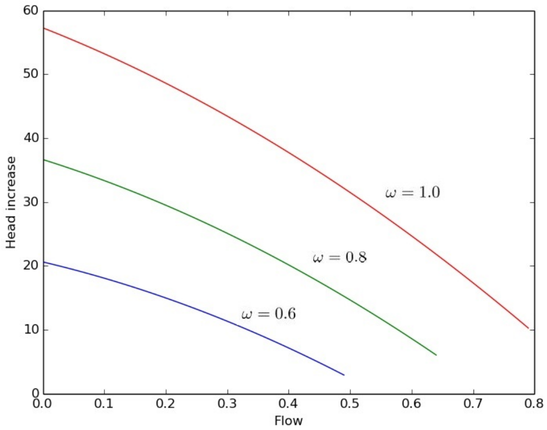


Figure 5.19: Example of characteristic curve for a pump with variable speed

Our solver SCIP can solve general nonconvex MIQCP [VG18; BHV12]. As a consequence, constraints consisting of any polynomial function can be handled by SCIP, e.g., by substituting them recursively until they contain only nonlinear terms in form of $x \cdot y$ or x^2. [1]

To enable that SCIP can handle the constraints like (5.22), we try to approximate function g with a polynomial function. Note that characteristic diagrams are usually given by the vendor with a set of measured points.

For polynomial fitting, on the one hand, we want to keep the degree of polynomials as low as possible. This is very helpful for the outer-approximation algorithms. On the other hand, the degree of polynomials should be high enough so that the approximation error is acceptable. For the computation we got a third real-world instance from Tsinghua University, Department of Hydraulic Engineering. Figure 5.20 shows a small water supply network `n9p3a11` in the suburbs of Beijing that contains 9 nodes (1 reservoir, 3 tanks, 5 junctions), 2 consumers, 3 pipes, 9 pumps, and 2 valves. Similar to the other 2 instances showed in Section 3.3.1, the network contains hourly demand forecast for one day.

All pumps in `n9p3a11` are variable speed pumps. With given data for the characteristic diagrams, we get approximated polynomials for energy consumption in form

$$C_{pt} = f_p(Q_{pt}, \omega_{pt}) = C_{0p}\omega_{pt}^3 + C_{1p}\omega_{pt}^2 Q_{pt} + C_{2p}\omega_{pt} Q_{pt}^2 + C_{3p} Q_{pt}^3 \tag{5.23}$$

with very acceptable approximation errors. Constants C_{0p}, C_{1p}, C_{2p} and C_{3p} are found by solving an NLP for every pump p.

For every constraint of the form $C_{pt} \geq f_p(Q_{pt}, \omega_{pt})$, variables Q_{pt} and ω_{pt} are box-constrained. The polynomial function $f_p(Q_{pt}, \omega_{pt})$ is exactly in the form of (5.17). Thus, Algorithm 5.2 and Algorithm 5.3 are applicable. For every of the four facets forming the box-constrained domain, we first compute the globally convex domain boundary points on that facet according to the

[1] Note that `SCIP` is already able to handle more complex nonlinear constraints directly, as explained in the latest user manual under http://scip.zib.de

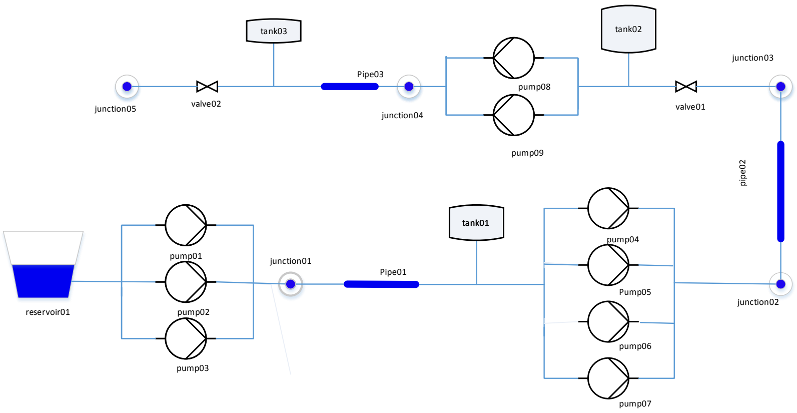


Figure 5.20: Schematic diagram of water network instance n9p3a11 with 9 nodes (1 reservoir, 3 tanks, 5 junctions), 2 consumers, 3 pipes, 9 pumps, and 2 valves.

algorithm introduced in the proof of Lemma 5.33. By case 2 of Lemma 5.33 we compute tight valid hyperplanes of type A based on point $x_i^{\min}, x_i^{\max}$ and $(x_i^{\min} + x_i^{\max})/2$. By case 3 we compute tight valid hyperplanes of type A based on points $x_i^{\min}$ and x_i^{mid}. By case 4 we compute tight valid hyperplanes of type A based on points x_i^{mid} and $x_i^{\max}$. In addition, since we have fixed four extreme points, two tight valid hyperplanes of type B are also computed. Finally, depending on the structure of the globally convex domain boundary points, six to eight linear constraints corresponding to tight valid hyperplanes are added for every constraint in the form of $C_{pt} \geq f_p(Q_{pt}, \omega_{pt})$. In the following we show a numerical example for n9p3a11.

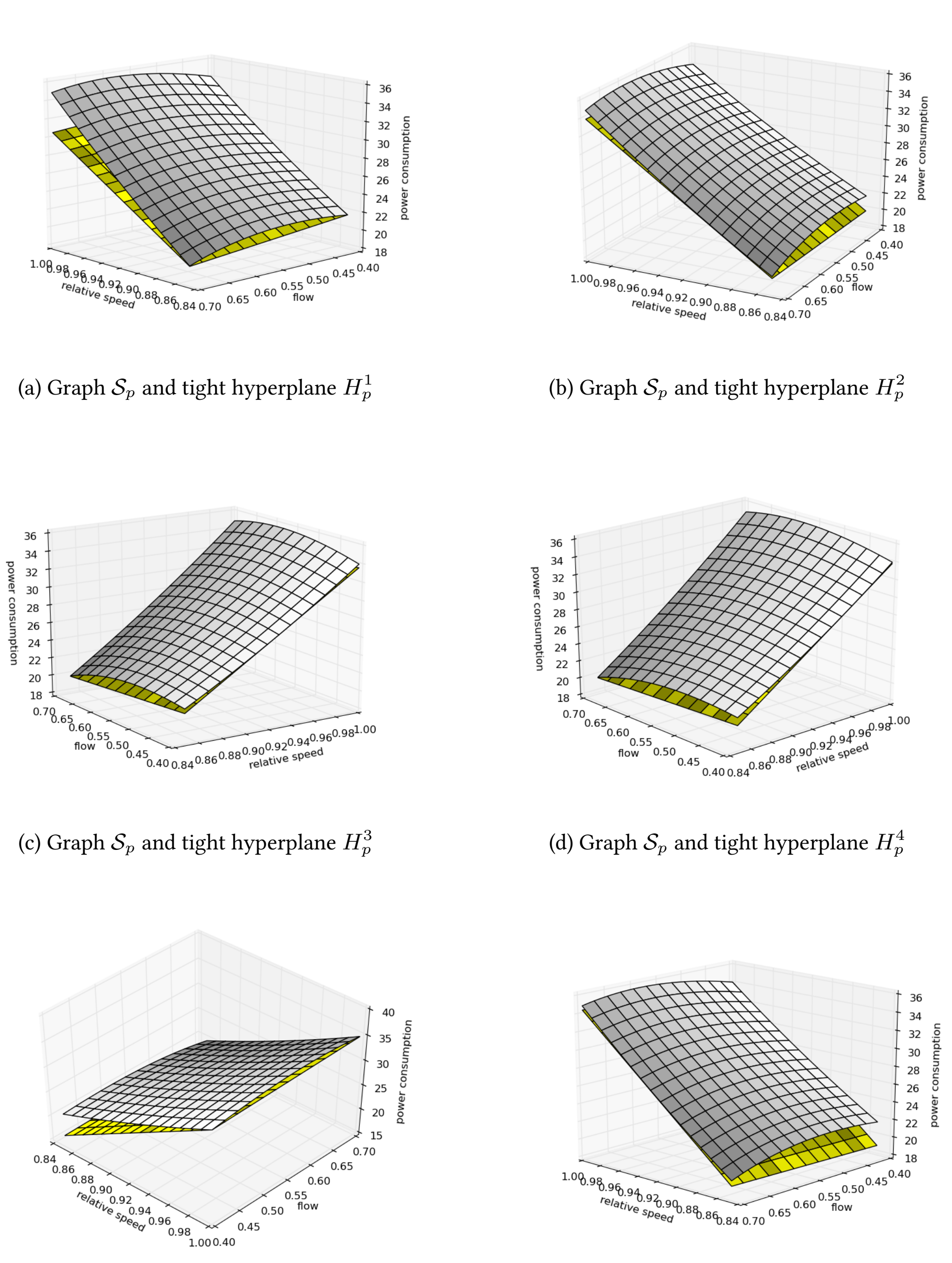
Example 5.41
In n9p3a11 there is a pump p with energy consumption constraint in the form of (5.23)

$$C_{pt} = f_p(Q_{pt}, \omega_{pt}) = 25.9267\,\omega_{pt}^3 + 18.1348\,\omega_{pt}^2 Q_{pt} + 22.1276\,\omega_{pt} Q_{pt}^2 - 42.6895\,Q_{pt}^3$$

with domain $X_p := \{(\omega_{pt}, Q_{pt}) \in [0.85, 1.0] \times [0.4, 0.7]\}$. The graph of $f_p(Q_{pt}, \omega_{pt})$ over X_p, denoted by $\mathcal{S}_p$ is shown as the white surface in Figure 5.21. Algorithms in Section 5.4 found six tight hyperplanes in the form of

$$H = \{(\omega_{pt}, Q_{pt}, C_{pt}) \mid C_{pt} = a\omega_{pt} + bQ_{pt} + c\},$$

where a, b, c are coefficients. The coefficients for the six tight hyperplanes are shown in Table 5.1. Note that the validity of each hyperplane has been verified by solving the corresponding NLP $(\text{OP}_{H^*}^{\min})$ at the end of Algorithm 5.2 and Algorithm 5.3, respectively. In addition, each tight hyperplane to $\mathcal{S}_p$ found above is shown as the yellow hyperplane in Figure 5.21, respectively.$\diamondsuit$



(a) Graph $\mathcal{S}_p$ and tight hyperplane H_p^1

(b) Graph $\mathcal{S}_p$ and tight hyperplane H_p^2

(c) Graph $\mathcal{S}_p$ and tight hyperplane H_p^3

(d) Graph $\mathcal{S}_p$ and tight hyperplane H_p^4

(e) Graph $\mathcal{S}_p$ and tight hyperplane H_p^5

(f) Graph $\mathcal{S}_p$ and tight hyperplane H_p^6

Figure 5.21: Original feasible region and linear underestimators in Example 5.41

Table 5.1: Tight hyperplanes to $\mathcal{S}_p$ and the corresponding coefficients in Example 5.41

Tight hyperplane	a	b	c
H_p^1	72.068325	-5.909471	-37.453828
H_p^2	95.828520	-0.365311	-61.693424
H_p^3	84.30486	-4.255828	-49.012439
H_p^4	88.619212	-2.912092	-53.620247
H_p^5	114.01154	2.774008	-81.132173
H_p^6	101.726218	1.355564	-68.27948

The experimental setup is the same as introduced in Section 4.4. Table 5.2 presents the computational results for the 24 original MINLPs and the corresponding MINLPs with extended pump cost relaxation constraints.

MINLP $P[0, i]$ denotes the operation problems for the first i hours with $i = 1, \dots, 24$. For every MINLP we set a time limit of one hour and gap limit with 10^{-5}. In column "(time) gap" the time is displayed if the gap limit has been reached within an hour. Otherwise the time limit has been reached, only the current gap needs to be displayed. Note that we only need to calculate the linear constraints one time before we start solving MINLP. This preprocessing only takes a few seconds in total. From the table we see that not only the dual bounds are improved significantly, most primal bounds are also improved for unsolved MINLP for operation of more than 14 hours.

Recall pump energy consumption constraint $C_{pt} \geq f_p(Q_{pt}, \omega_{pt})$, variable C_{pt} is contained in the objective. Better outer-approximation as well as tighter relaxation will improve the dual bound directly. Our computational results have verified it. A graphic comparison of the primal and dual bounds is shown in Figure 5.22.

Table 5.2: Detailed computational results for MINLPs with extended pump cost relaxation constraints and the original MINLPs for `n9p3a11`, computed by SCIP 5.0.1

MINLP	original			with ext. pump cost rel. cons.		
	primal	dual	(time) gap	primal	dual	(time) gap
P[0, 1]	19.46	19.46	(0.48) 0	19.38	19.38	(0.42) 0
P[0, 2]	39.57	39.57	(1.38) 0	39.5	39.5	(1.22) 0
P[0, 3]	215	215	(1857.85) 0	215	215	(260.7) 0
P[0, 4]	247.63	247.63	(2797.98) 0	247.63	247.63	(1366.6) 0
P[0, 5]	324.33	214.13	0.514659	314.94	314.8	0.000470589
P[0, 6]	456	163.65	1.78649	404.81	343.41	0.178792
P[0, 7]	586.42	237.08	1.47354	586.28	407.72	0.437932
P[0, 8]	822.62	253.84	2.24071	782.35	558.13	0.401734
P[0, 9]	1113.14	238.28	3.67158	1115	727.98	0.531645
P[0, 10]	1494.42	423.74	2.52677	1518.81	940.26	0.615304
P[0, 11]	1914.45	379.87	4.03972	1934.06	1146.68	0.686657
P[0, 12]	2603.85	461.55	4.64152	2669.16	1451.5	0.838896
P[0, 13]	3174.84	546.05	4.81419	3197.05	1690.12	0.891608
P[0, 14]	4127.24	442.67	8.3235	3674.65	1882.28	0.952236
P[0, 15]	4373.83	526.89	7.30122	4056.38	2073.05	0.956725
P[0, 16]	4806.56	643.72	6.46681	4757.25	2354.57	1.02043
P[0, 17]	5190.04	736.59	6.04607	5023.09	2616.48	0.919789
P[0, 18]	5909.72	504.07	10.724	5636.12	2970.92	0.897098
P[0, 19]	6731.29	575.39	10.6987	6393.88	3071.39	1.08175
P[0, 20]	7491.98	642.96	10.6524	7014.98	3543.4	0.979729
P[0, 21]	7831.22	476.56	15.4329	7439.11	3636.75	1.04554
P[0, 22]	8785.58	523.11	15.7948	7998.55	3823.91	1.09172
P[0, 23]	8147.78	571.52	13.2562	8124.75	3791.95	1.14263
P[0, 24]	8380.79	454.64	17.434	8271.19	4070.3	1.03208

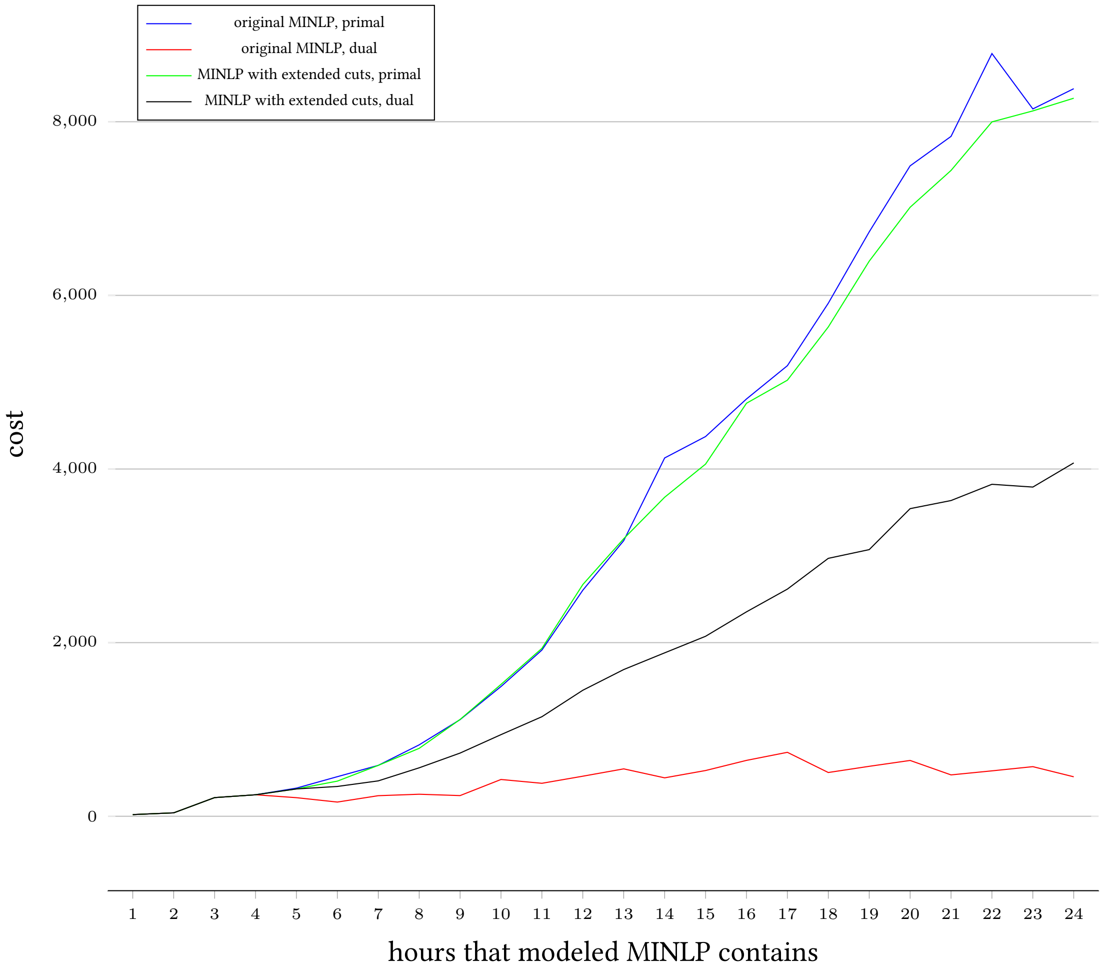


Figure 5.22: Comparison of primal and dual bounds by solving MINLPs with extended pump cost relaxation constraints and the original MINLPs for `n9p3a11`

Chapter 6

Conclusion and Outlook

This thesis deals with planning problems for the optimal operation of water supply networks. At the beginning, accompanying a literature survey we argue why we use the MINLP approach to model and to solve the problem.

Our contribution starts from Chapter 2. In the first part we present a detailed MINLP model. During the verification of computed solutions by our solver we detect modeling errors that cannot be found in any other literature. For that we add additional variables and constraints to correct the model. In the first part we present several reformulation and presolving techniques. These techniques can be utilized once the corresponding network properties are detected.

For such a nonconvex MINLP problem, there are plenty of works for solutions to local optimality. Our final goal is to reach ε-global optimality since the solution quality is expected to be verified. In Chapter 3 we reduce the dynamic version of the operation problem for 24 hours to a version of a fixed point of time. This helps us to verify the model and the data of instances.

The first major academic contributions of the thesis appear in Chapter 4. There we work on subnetworks that only contain pipes and junctions. These subnetworks appear very commonly since nowadays network design pipes that are contained in cycles are used as backups. This is necessary if a part of the subnetworks have technical interruptions. For a general subnetwork defined there, we prove that the corresponding nonlinear constraints which consist of a nonlinear feasibility problem can always be uniquely solved. Based on the derived theorem, we show that the unique solvability is reached by symbolic computation for our instance using Maple [Map]. However, the exact symbolic solution cannot be used by SCIP directly. Instead, we approximate the functions for the solution above with polynomial fitting. The approximated polynomial function can be handled by SCIP and has very little approximative error. The simplified MINLPs are easier to solve, proved by computational results.

Although the exact symbolic solution cannot help SCIP directly, it can be very useful e.g., for developing simulation tools. The property of unique solvability may also contribute to develop algorithms to check feasibility of the entire network.

The second major scientific contribution of the thesis can be found in Chapter 5. The idea comes from the very early computational results for instance n9p3a11, presented in Section 5.5. Even the instance is much smaller and structurally easier than n88p64a64 and n25p22a18, primal solutions cannot always be found rapidly and the dual bounds are also very “bad”. The only difference is that the pumps in this instance can be operated with a variable speed. Due to one more dimension in the characteristic curve, nonconvexities are increased both in the constraints and especially in the objective. Hence the low quality of dual bounds is expectable. During seeking a method to generate better outer-approximation for the nonlinear terms, we found very little literature which can help us. As a consequence, we decide to investigate the description of the convex hull of graphs of polynomial functions over a polytope in the thesis, in particular for dimension n.

An additional remark is that we only generate additional linear constraints (we call "valid 3D cuts") before we start solving MINLP. Note that during the branch-and-bound approach the new subproblem contains changed and tighter variable bounds, new valid 3D cuts can be generated adaptively and integrated into the MINLP solving process. We prefer to leave the implementation to be done for real industrial requirement rather than in this research thesis.

At the end, in addition to the promising results presented in this thesis, we think there are still a lot of topics we can continue the research. First of all, we need better primal solutions. Instead of using general MINLP heuristics, we can try to find special heuristics for our certain problem. Furthermore, according to our computational results, there is still much room for the improvement of dual bounds to verify the quality of solutions. Besides the idea of adaptive refinement of out-approximation of polynomial constraints, decomposition techniques based on Lagrangian relaxation should also help us to improve the dual bounds.

List of Figures

All figures in this thesis were created by the author. For most of the figures, the wonderful Python libs and TEX packages TikZ and PGF by Till Tantau [Tan08] were used.

List of Tables

List of Algorithms

a

Bibliography

[AB06] C. D. Aliprantis and K. C. Border. *Infinite Dimensional Analysis*. Berlin: Springer, 2006.

[AB10] Kurt M. Anstreicher and Samuel Burer. “Computable representations for convex hulls of low-dimensional quadratic forms”. In: *Math. Program* 124 (1-2), 2010, pp. 33–43.

[Ach07] T. Achterberg. “Constraint Integer Programming”. PhD thesis. Technische Universität Berlin, 2007.

[Ach09] T. Achterberg. “SCIP: solving constraint integer programs”. English. In: *Mathematical Programming Computation* 1 (1), 2009, pp. 1–41. DOI: 10.1007/s12532-008-0001-1.

[ALL06] Kumar Abhishek, Sven Leyffer, and J. T. Linderoth. “FilMINT: An Outer-Approximation-Based Solver for Nonlinear Mixed Integer Programs”. In: *Argonne National Laboratory, Mathematics and Computer Science Division*, 2006.

[AM95] I. P. Androulakis and C. D. Maranas. *alphaBB: A Global Optimization Method for General Constrained Nonconvex Problems.* en. 1995.

[BBCCGLLLMSW08] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. “An algorithmic framework for convex mixed integer nonlinear programs”. In: *Discrete Optimization* 5 (2), 2008, pp. 186–204.

[BD56] Garrett Birkhoff and J. B. Diaz. “Non-linear network problems”. In: *Quarterly of Applied Mathematics* 13 (4), 1956, pp. 431–443. DOI: 10.1090/qam/77398.

[BDKMR17] Natashia Boland, Santanu S. Dey, Thomas Kalinowski, Marco Molinaro, and Fabian Rigterink. “Bounding the gap between the McCormick relaxation and the convex hull for bilinear functions”. In: *Math. Program* 162 (1-2), 2017, pp. 523–535.

[BDLLT06] Cristiana Bragalli, Claudia D'Ambrosio, Jon Lee, Andrea Lodi, and Paolo Toth. "An MINLP Solution Method for a Water Network Problem". In: *Algorithms - ESA 2006, 14th Annual European Symposium, Zurich, Switzerland, September 11-13, 2006, Proceedings*. Ed. by Yossi Azar and Thomas Erlebach. Vol. 4168. Lecture Notes in Computer Science. Springer, 2006, pp. 696–707.

[BDLLT08] Cristiana Bragalli, Claudia D'Ambrosio, Jon Lee, Andrea Lodi, and Paolo Toth. *Water Network Design by MINLP*. Tech. rep. RC24495. IBM, 2008.

[BDLLT12] Cristiana Bragalli, Claudia DâAmbrosio, Jon Lee, Andrea Lodi, and Paolo Toth. "On the optimal design of water distribution networks: a practical MINLP approach". In: *Optimization and Engineering* 13, 2 2012, pp. 219–246. DOI: 10.1007/s11081-011-9141-7.

[Ber06] T. Berthold. "Primal Heuristics for Mixed Integer Programs". MA thesis. Technische Universität Berlin, 2006.

[Ber09] D. P. Bertsekas. *Convex Optimization Theory*. Athena Scientific, 2009.

[Ber14] Timo Berthold. "Heuristic algorithms in global MINLP solvers". PhD thesis. 2014, p. 366.

[BF76] E. M. L. Beale and J. J. H. Forrest. "Global optimization using special ordered sets". In: *Mathematical Programming* 10, 1976, pp. 52–59.

[BG10] Timo Berthold and Ambros M. Gleixner. "Undercover – a primal heuristic for MINLP based on sub-MIPs generated by set covering". In: *Proceedings of the EWMINLP*. Ed. by Pierre Bonami, Leo Liberti, Andrew J. Miller, and Annick Sartenaer. 2010, pp. 103–112.

[BG12] Timo Berthold and Ambros M. Gleixner. *Undercover – a primal MINLP heuristic exploring a largest sub-MIP*. ZIB-Report 12-07. http://vs24.kobv.de/opus4-zib/frontdoor/index/index/docId/1463/. Zuse Institute Berlin, 2012.

[BGS04] Jens Burgschweiger, Bernd Gnädig, and Marc C. Steinbach. *Optimization Models for Operative Planning in Drinking Water Networks*. eng. ZIB-Report 04-48. http://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/823. Zuse Institute Berlin, 2004.

[BGS05] Jens Burgschweiger, Bernd Gnaedig, and Marc Steinbach. "Nonlinear Programming Techniques for Operative Planning in Large Drinking Water Networks". In: *ZIB-Report* 05-31, 2005.

[BHPV11] Timo Berthold, Stefan Heinz, Marc E. Pfetsch, and Stefan Vigerske. "Large Neighborhood Search beyond MIP". In: *Proceedings of the 9th Metaheuristics International Conference (MIC 2011).* Ed. by Luca Di Gaspero, Andrea Schaerf, and Thomas Stützle. 2011, pp. 51–60.

[BHV12] Timo Berthold, Stefan Heinz, and Stefan Vigerske. "Extending a CIP framework to solve MIQCPs". In: *Mixed-integer nonlinear optimization.* Ed. by Jon Lee and Sven Leyffer. Vol. 154. The IMA volumes in Mathematics and its Applications. Springer, 2012, pp. 427–444. DOI: 10.1007/978-1-4614-1927-3_15.

[BLLMW09] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wachter. "Branching and bounds tightening techniques for non-convex MINLP". In: *Optimization Methods and Software* 24 (4–5), 2009, pp. 597–634.

[BMA14] Jacek Blaszczyk, Krzysztof Malinowski, and Alnoor Allidina. "Optimal Pump Scheduling By NLP For Large Scale Water Transmission System". In: *28th European Conference on Modelling and Simulation, ECMS 2014, Brescia, Italy, May 27-30, 2014.* Ed. by Flaminio Squazzoni, Fabio Baronio, Claudia Archetti, and Marco Castellani. European Council for Modeling and Simulation, 2014, pp. 501–507.

[BMV13] Martin Ballerstein, Dennis Michaels, and Stefan Vigerske. *Linear Underestimators for bivariate functions with a fixed convexity behavior.* eng. Tech. rep. 13-02. Takustr. 7, 14195 Berlin: ZIB, 2013.

[BV10] Michael R. Bussieck and Stefan Vigerske. "MINLP Solver Software". 2010.

[BZBY08] Mario T. L. Barros, Renato C. Zambon, Paulo S. F. Barbosa, and William W.-G. Yeh. "Planning and Operation of Large-Scale Water Distribution Systems with Preemptive Priorities". In: *Journal of Water Resources Planning and Management.* Vol. 134. May 2008.

[CCVHLKJL78] M Collins, L Cooper, R V. Helgason, J L. Kennington, and L J. Leblanc. "Solving the Pipe Network Analysis Problem Using Optimization Techniques". In: *Management Science.* Vol. 24. Mar. 1978, pp. 747–760.

[CLO92] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms.* Springer-Verlag, 1992.

[Cpp] *CppAD. A Package for Differentiation of C++ algorithms.* http://www.coin-or.org/CppAD/.

[DG86] M. A. Duran and I. E. Grossman. "An outer-approximation algorithm for a class of mixed-integer nonlinear programs". In: *Mathematical Programming* 36, 1986, pp. 307–339.

[DGKLMM11] Pia Domschke, Björn Geißler, Oliver Kolb, Jens Lang, Alexander Martin, and Antonio Morsi. "Combination of Nonlinear and Linear Optimization of Transient Gas Networks". In: *INFORMS Journal on Computing* 23 (4), 2011, pp. 605–617.

[DGPS12] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann. "Singular 3-1-6 — A computer algebra system for polynomial computations". http://www.singular.uni-kl.de. 2012.

[DKL10] Pia Domschke, Oliver Kolb, and Jens Lang. "An adaptive model switching and discretization algorithm for gas flow on networks". In: *Procedia Computer Science* 1 (1), May 2010, pp. 1331–1340.

[DKL15] Pia Domschke, Oliver Kolb, and Jens Lang. "Adjoint-based error control for the simulation and optimization of gas and water supply networks". In: *Applied Mathematics and Computation* 259, 2015, pp. 1003–1018.

[DLMY95] Ali Diba, Peter W. F. Louie, Manouchehr Mahjoub, and William W-G. Yeh. "Planned Operation of Large-Scale Water-Distribution System". In: *Water Resources Planning and Management* 121, 3 1995.

[Edw94] C. H. Edwards. *Advanced calculus of several variables / C.H. Edwards, Jr.* New York: Dover Publications, INC., 1994.

[Ehr05] Matthias Ehrgott. *Multicriteria Optimization (2. ed.)* Springer, 2005, pp. I–XIII, 1–323.

[Epa] *EPANET. A software that models water distribution piping systems.* `http://www.epa.gov/nrmrl/wswrd/dw/epanet.html` (accessed March 2012).

[FL94] Roger Fletcher and Sven Leyffer. "Solving mixed integer nonlinear programs by outer approximation". In: *Math. Program* 66, 1994, pp. 327–349.

[FT07] J.J.H. Forrest and J.A. Tomlin. "Branch and bound, integer, and non-integer programming". In: *Annals of Operations Research* 149 (1), 2007, pp. 81–87.

[FT15] Dariush Fooladivanda and Joshua A. Taylor. "Optimal pump scheduling and water flow in water distribution networks". In: *CDC*. IEEE, 2015, pp. 5265–5271.

[GDS94] Johannes Grotendorst, Juergen Dornseiffer, and Siegfried M. Schoberth. *Symbolic-numeric computation techniques for problem-solving in Physical Chemistry and Biochemistry.* Tech. rep. KFA-ZAM-IB-9414. Jülich, Germany: KFA Research Centre, Jülich, 1994, p. 9.

[GHHV12] Ambros Gleixner, Harald Held, Wei Huang, and Stefan Vigerske. "Towards Globally Optimal Operation of Water Supply Networks". In: *Numerical Algebra, Control and Optimization* 2 (4), 2012, pp. 695–711.

[GK97] I.E. Grossmann and Z. Kravanja. "Mixed-integer nonlinear programming: A survey of algorithms and applications". 1997.

[GKLLMM11] Björn Geißler, Oliver Kolb, Jens Lang, Günter Leugering, Alexander Martin, and Antonio Morsi. "Mixed integer linear models for the optimization of dynamical transport networks". In: *Mathematical Methods of Operations Research* 73, 3 2011, pp. 339–362. DOI: 10.1007/s00186-011-0354-5.

[GKZ90] V. P. Gerdt, N. V. Khutornoy, and A. Yu. Zharkov. "Solving algebraic systems which arise as necessary integrability conditions for polynomial-nonlinear evolution equations". In: *ISSAC '90: proceedings of the International Symposium on Symbolic and Algebraic Computation: August 20–24, 1990, Tokyo, Japan.* Ed. by Shunro Watanabe and Morio Nagata. pub-ACM:adr and pub-AW:adr: ACM Press and Addison-Wesley, 1990, pp. 299–299.

[Gle15] Ambros M. Gleixner. "Exact and fast algorithms for mixed-integer nonlinear programming". PhD thesis. Technische Universität Berlin, 2015.

[GPSSS17] Martin Gross, Marc E. Pfetsch, Lars Schewe, Martin Schmidt, and Martin Skutella. "Algorithmic Results for Potential-Based Flows: Easy and Hard Cases". 2017.

[GVVRK02] Ignacio E. Grossmann, Jagadisan Viswanathan, Aldo Vecchietti, Ramesh Raman, and Erwin Kalvelagen. *GAMS/DICOPT: A Discrete Continuous Optimization Package.* en. 2002.

[Hae08] C. Haehnlein. "Numerische Modellierung zur Betriebsoptimierung von Wasserverteilnetzen". PhD thesis. Technische Universität Darmstadt, 2008.

[Hua11] Wei Huang. "Operative Planning of Water Supply Networks by Mixed Integer Nonlinear Programming". MA thesis. Freie Universität Berlin, 2011.

[HUL01] Jean-Baptiste Hiriart-Urruty and Claude Lemarechal. *Fundamentals of Convex Analysis.* Jan. 2001, pp. 163–208.

[Ipo] *Ipopt. Interior Point Optimizer.* http://www.coin-or.org/Ipopt/.

[JMW08] Matthias Jach, Dennis Michaels, and Robert Weismantel. "The Convex Envelope of (n−1)-Convex Functions". In: *SIAM Journal on Optimization* 19 (3), 2008, pp. 1451–1466.

[Kar84] N. Karmarkar. “A New Polynomial–Time Algorithm for Linear Programming”. In: *Combinatorica* 4, 1984, pp. 373–395.

[Kha79] L. G. Khachian. “A polynomial time algorithm for linear programming”. In: *Soviet Math. Dokl.* 20, 1979, pp. 191–194.

[KLB10] Oliver Kolb, Jens Lang, and Pia Bales. “An implicit box scheme for subsonic compressible flow with dissipative source term”. In: *Numerical Algorithms* 53 (2-3), 2010, pp. 293–307.

[KLLMMOOR12] Katrin Klamroth, Jens Lang, Günter Leugering, Alexander Martin, Antonio Morsi, Martin Oberlack, Manfred Ostrowski, and Roland Rosen. *Mathematical Optimization of Water Networks.* Vol. 162. International Series of Numerical Mathematics. Basel: Birkhäuser-Science, 2012.

[Kol11] Oliver Kolb. “Simulation and Optimization of Gas and Water Supply Networks”. PhD thesis. Technische Universität Darmstadt, 2011.

[KS12] Aida Khajavirad and Nikolaos V. Sahinidis. “Convex envelopes of products of convex and component-wise concave functions”. In: *J. Global Optimization* 52 (3), 2012, pp. 391–409.

[KS13] Aida Khajavirad and Nikolaos V. Sahinidis. “Convex envelopes generated from finitely many compact convex sets”. In: *Math. Program* 137 (1-2), 2013, pp. 371–408.

[Laz83] D. Lazard. “Gröbner Bases, Gaussian Elimination and Resolution of Systems of Algebraic Equations”. In: *Proc. European Computer Algebra Conference, EUROCAL ’83, LNCS 162.* Ed. by J. A. van Hulzen. London: Springer Verlag, 1983, pp. 146–156.

[LD60] Ailsa H. Land and Alison G. Doig. “An automatic method for solving discrete programming problems”. In: *Econometrica* 28, 1960, pp. 497–520.

[Ley01] S. Leyffer. “Integrating SQP and branch-and-bound for mixed integer nonlinear programming”. In: *Computational Optimization and Applications* 18, 2001, pp. 295–309.

[Loc15] Marco Locatelli. “Convex Envelopes of Some Quadratic Functions over the n-Dimensional Unit Simplex”. In: *SIAM Journal on Optimization* 25 (1), 2015, pp. 589–621.

[Loc16] Marco Locatelli. “Polyhedral subdivisions and functional forms for the convex envelopes of bilinear, fractional and other bivariate functions over general polytopes”. In: *J. Global Optimization* 66 (4), 2016, pp. 629–668.

[LP03] Leo Liberti and Constantinos C. Pantelides. “Convex Envelopes of Monomials of Odd Degree”. In: *J. of Global Optimization* 25, 2 2003, pp. 157–168. DOI: 10.1023/A:1021924706467.

[LS09] Youdong Lin and Linus Schrage. “The global solver in the LINDO API”. In: *Optimization Methods & Software* 24 (4–5), 2009, pp. 657–668. DOI: 10.1080/10556780902753221.

[LS13] Marco Locatelli and Fabio Schoen. *Global Optimization: Theory, Algorithms, and Applications.* 2013, p. 437.

[LS14] Marco Locatelli and Fabio Schoen. “On convex envelopes for bivariate functions over polytopes”. In: *Math. Program* 144 (1-2), 2014, pp. 65–91.

[LW01] Jon Lee and Dan Wilson. “Polyhedral methods for piecewise-linear functions I: the lambda method”. In: *Discrete Applied Mathematics* 108 (3), 2001, pp. 269–285.

[LW09] Andreas Lundell and Tapio Westerlund. “Convex underestimation strategies for signomial functions”. In: *Optimization Methods and Software* 24 (4-5), 2009, pp. 505–522.

[Map] *Maple.* http://www.maplesoft.com. Waterloo, Ontario, Canada: Waterloo Maple Inc.

[Mau77] J. J. Maugis. *Étude de réseaux de transport et de distribution de fluide.* 1977.

[MF04] Clifford A. Meyer and Christodoulos A. Floudas. “Trilinear Monomials with Mixed Sign Domains: Facets of the Convex and Concave Envelopes”. In: *J. Global Optimization* 29 (2), 2004, pp. 125–155.

[MF05] C. A. Meyer and C. A. Floudas. “Convex envelopes for edge-concave functions”. In: *Mathematical Programming* 103 (2), 2005, pp. 207–224.

[MG94] M. F. Mahmood and T. L. Gill. “Analytical Approaches to Solving Coupled Nonlinear Schrödinger Equations Using Maple V”. In: *Maple V: mathematics and its application: proceedings of the Maple Summer Workshop and Symposium, Rensselaer Polytechnic Institute, Troy, New York, August 9–13, 1994.* Ed. by R. J. Lopez. pub-BIRKHAUSER:adr: Birkhäuser, 1994, pp. 83–89.

[MMM06] Alexander Martin, Markus Möller, and Susanne Moritz. “Mixed Integer Models for the Stationary Case of Gas Network Optimization”. In: *Mathematical Programming, Ser. B* 105, 2006. doi:10.1007/s10107-005-0665-5, pp. 563–582.

[Mor13] Antonio Morsi. “Solving MINLPs on Loosely-Coupled Networks with Applications in Water and Gas Network Optimization”. PhD thesis. Universität Erlangen-Nürnberg, 2013.

[Oai] *GAMS - The Solver Manuals*. GAMS Development Corp. Washington DC, 2009.

[PJNN14] Louise Brac de la Perrière, Antoine Jouglet, Alexandre Nace, and Dritan Nace. *Water planning and management: An Extended Model for the Real-Time Pump Scheduling Problem*. HAL CCSD, 2014.

[RB09] M. Roelofs and J. Bisschop. *AIMMS 3.9 - The Language Reference*. Paragon Decision Technology B.V. Haarlem, The Netherlands, 2009.

[RlRmL14] Eugenio Roanes-lozano, Eugenio Roanes-macías, and Luis M. Laita. *Some applications of Gröbner bases*. en. 2014.

[RMWSB02] Roger Z. Ríos-Mercado, Suming Wu, L. Ridgway Scott, and E. Andrew Boyd. “A Reduction Technique for Natural Gas Transmission Network Optimization Problems”. In: *Annals OR* 117 (1-4), 2002, pp. 217–234.

[Roc70] R. T. Rockafellar. *Convex analysis*. Princeton University Press, 1970.

[Sah96] N. Sahinidis. “BARON: A general purpose global optimization software package”. In: *Journal of Global Optimization* 8, 1996, pp. 201–205.

[Sch03] A. Schrijver. *Combinatorial optimization : polyhedra and efficiency*. Springer, 2003.

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, 1986.

[Sci] *SCIP. Solving Constraint Integer Programs*. `http://scip.zib.de/`.

[She97] Hanif Sherali. “Convex envelopes of multilinear functions over a unit hypercube and over special discrete sets”. In: *Acta Mathematica Vietnamica*. Vol. 22. Jan. 1997.

[Sko11] A. Skopenkov. *A simple proof of the Abel-Ruffini theorem*. ru. Comment: 10 pages, 3 figures, in Russian. references updated. 2011.

[SM00] A. Burcu Altan Sakarya and Larry W. Mays. “Optimal Operation of Water Distribution Pumps Considering Water Quality”. In: *Journal of Water Resources Planning and Management*. Vol. 126. July 2000.

[Sop] *SoPlex. Sequential object-oriented simPlex*. `http://soplex.zib.de/`.

[SP99] E. M. B. Smith and C. C. Pantelides. “A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs”. In: *Computers & Chemical Engineering* 23, 1999, pp. 457–478.

[Tan08] Till Tantau. *The TikZ and PGF Packages. Manual for version 2.00.* Lübeck, Feb. 2008.

[TRX13] Mohit Tawarmalani, Jean-Philippe P. Richard, and Chuanhui Xiong. "Explicit convex and concave envelopes through polyhedral subdivisions". In: *Math. Program* 138 (1-2), 2013, pp. 531–577.

[TS02a] M. Tawarmalani and N. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications.* Dordrecht, The Netherlands: Kluwer, 2002.

[TS02b] Mohit Tawarmalani and Nikolaos V. Sahinidis. "Convex extensions and envelopes of lower semi-continuous functions". In: *Math. Program* 93 (2), 2002, pp. 247–263.

[TS04] Mohit Tawarmalani and Nikolaos V. Sahinidis. "Global optimization of mixed-integer nonlinear programs: A theoretical and computational study". In: *Mathematical Programming, Ser. A* 99, 2004. doi:10.1007/s10107-003-0467-6, pp. 563–591.

[TS05] M. Tawarmalani and N. Sahinidis. "A polyhedral branch-and-cut approach to global optimization". In: *Mathematical Programming* 103 (2), 2005, pp. 225–249.

[VG18] Stefan Vigerske and Ambros M. Gleixner. "SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework". In: *Optimization Methods and Software* 33 (3), 2018, pp. 563–593.

[Vig12] Stefan Vigerske. "Decomposition of Multistage Stochastic Programs and a Constraint Integer Programming Approach to Mixed-Integer Nonlinear Programming". PhD thesis. Humboldt-Universität zu Berlin, 2012.

[Wal03] T.M. Walski. *Advanced water distribution modeling and management.* v. 1. Haestead Press, 2003.

[WB06] Andreas Wächter and Lorenz T. Biegler. "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming". In: *Mathematical Programming* 106 (1), 2006, pp. 25–57. DOI: 10.1007/s10107-004-0559-y.

[WK07] Donald V. Chase-Dragan Savic Walter Grayman Stephanus Beckwith Walski Thomas M and E Koelle. *Advanced Water Distribution Modeling and Management.* Bentley Institute Press, 2007.

[WP02] T. Westerlund and R. Pörn. "Solving Pseudo-Convex Mixed Integer Optimization Problems by Cutting Plane Techniques". In: *Optimization and Engineering* 3 (3), 2002, pp. 253–280.

[WP95] T. Westerlund and F. Pettersson. “An Extended Cutting Plane Method for Solving Convex MINLP Problems”. In: *Computers & Chemical Engineering* 19, 1995, S131–S136.

[WS73] Wynn R. Walker and Gaylord V. Skogerboe. “Mathematical Modeling of Water Management Strategies in Urbanizing River Basins”. In: *Colorado Water Resources Research Institute*, 1973.

[Zac96] Joseph Zachary. *Introduction to scientific programming: computational problem solving using Maple and C.* pub-SV:adr: Springer-Verlag, 1996, pp. xxiv + 380. DOI: `https://doi.org/10.1007/978-1-4612-2366-5`.

[Zim] *Zimpl. Zuse Institute Mathematical Programming Language.* `http://zimpl.zib.de/`.

[ZS89] U Zessler and Uri Shamir. “Optimal Operation of Water Distribution Systems”. In: *Journal of Water Resources Planning and Management.* Vol. 115. Nov. 1989.

Academic CV of Wei Huang

2019	PhD in Mathematics
Dec. 2017 - 2019	PhD Student at the Graduate School Computational Engineering of the Technische Universität Darmstadt
2011 - Nov. 2017	PhD Student at the TUM Graduate School of the Technische Universität München
2011	Diploma (MSc) in Mathematics and Diploma (MSc) in Computer Science, Freie Universität Berlin
2006 - 2011	Studies of Mathematics und Computer Science at the Freie Universität Berlin