Chapter 1

Introduction

The work presented in the following was motivated by an ongoing cooperation between the *Konrad-Zuse-Zentrum für Informationstechnik* and the *DFN-Verein zur Förderung eines Deutschen Forschungsnetzes e.V.*. The DFN-Verein is a non-profit organization established to promote computer-based communication and information services for research, development, and education in Germany. Among other activities, the DFN-Verein operates the *Deutsches Forschungsnetz* DFN, which is Germany's national research and education network. Connecting universities, research centers, schools, libraries and other institutions from all over Germany, it provides high-performance communication infrastructure for the German research and education community. Being connected to the global Internet and to the European backbone network GÉANT, the DFN is an integral part of the worldwide system of research and education networks. Between 2000 and 2006 the national backbone of the DFN was the *Gigabit Wissenschaftsnetz* G-WiN, since 2006 it is the so-called X-WiN.

One of the main tasks of the DFN is to provide IP connectivity of proven high quality among the participating institutions. The network must not only be able to handle the large data streams arising in scientific applications, it must also meet the high Quality-of-Service standards required for multimedia applications such as video lectures or video conferences. For this reason, G-WiN and X-WiN have been implemented as (virtual) private networks and are managed directly by the DFN-Verein.

1.1 Focus of this Thesis

In this thesis, we investigate optimization problems that arise in the planning and operation of IP networks such as G-WiN or X-WiN. In principle, these are the same problems that arise in planning of other communication networks: In the very long-term strategic network planning, the network provider or operator must decide about the future node locations, network hierarchies if necessary, and transmission technologies. In long- to mid-term

planning, the provider typically wishes to find an optimal (re)design of the network. This involves decisions concerning the network topology, the hardware and capacity installation, and the routing of the traffic demands. The goal of these long-term problems usually is to minimize (an estimation of) the total network cost. Finally, in the short-term operational planning, the network operator's goal is to make the best possible use of the available network resources. Usually, this means to reconfigure the traffic flows according to performance objectives – often in response to traffic demand changes – but leaving the network's topology and hardware configuration unchanged. In practice, this task is called traffic engineering.

A particular issue in IP networks is the way the traffic is routed through the network and the way this routing can be controlled by the network operator. Frankly, each data packet is sent along a shortest path towards its destination. Depending on the technical implementation of the routing protocol, all packets sent from one origin to one destination have to follow the same path or may be distributed among all shortest paths. The only mechanism to control this routing is to modify the metric that is used to compute the shortest paths. The main advantages and reasons for the popularity of this routing scheme are its simplicity and its robustness. It can be implemented in a distributed fashion, it is inherently robust against network failures, and it requires no centralized routing management. Because packet forwarding decision depend only on the destination address, it also scales much better with the network size than routing schemes that are based on pre-configured end-to-end paths. On the other hand, shortest path routing is less bandwidth efficient than other routing schemes and it is extremely complicated from the network planning perspective. Because all routing paths are based on the same shortest path metric, the attempt to change one end-to-end routing path by fiddling around with this metric will affect other routing paths, too. In contrast to many other routing schemes, there are strong and rather complicated interdependencies among the end-to-end paths that comprise a valid routing. Therefore, the routing paths in an IP network can be controlled and modified only together as a whole. Finding a metric that induces a set of globally efficient paths is one of the most important and most difficult problems in IP networks.

In this thesis, we consider the unsplittable shortest path routing variant. In this shortest path routing version, each traffic demand shall be sent unsplit via a single path from its origin to its destination. Accordingly, the metric must be chosen such that the shortest paths are uniquely determined for all demand node pairs. This routing version is used in the G-WiN and in the X-WiN.

1.2 Contributions of this Thesis

The main contributions in this thesis can be summarized as follows:

1. We prove that the problem of finding an integer-valued metric that induces a prescribed set of unique shortest paths and minimizes the longest arc or the longest path length is $\mathcal{APX}$-hard. Previously, it was even open if these problems are $\mathcal{NP}$-hard or not. The proof is given Chapter 4.

2. We introduce an independence system characterization for unsplittable shortest path routings. For every digraph, the family of all path sets that comprise an unsplittable shortest path routing forms an independence system. The circuits in this independence system are inclusion-wise minimal path sets that cannot be realized as an unsplittable shortest path routing. We present a simple greedy algorithm that finds such an inclusion-wise minimal conflict in a given path set that is not an unsplittable shortest path routing in polynomial time. Furthermore, we show that the problem of finding a minimum cardinality or minimum weight such conflict is $\mathcal{NP}$-hard to approximate within a constant factor less than 7/6. Analogous results are shown for an alternative independence system characterization of unsplittable shortest path routings by so-called forwardings. The two independence system characterizations and the related results are contained in Chapter 5.

3. We thoroughly analyze the computational complexity of the basic network design and traffic engineering problems with unsplittable shortest path routing in Chapter 6. We show that, for a given capacitated digraph and a given set of commodities, the minimal congestion that is achievable with unsplittable shortest path routing may be a factor of $\Omega(|V|^2)$ larger than the minimum congestion that is achievable with unsplittable flow routing, with shortest multi-path routing, or with fractional multicommodity flow routing in general. We also prove several inapproximability results for unsplittable shortest path routing problems that are harder than the best known results for the corresponding unsplittable flow problems. For example, we show that it is $\mathcal{NP}$-hard to a approximate the minimum congestion that is is achievable with unsplittable shortest path routing within a factor of $\mathcal{O}(|V|^{1-\epsilon})$ for any $\epsilon > 0$. Several polynomial time approximation algorithms are discussed as well.

4. We develop a practically useful mixed-integer linear programming approach to solve real-world network design and traffic engineering problems with unsplittable shortest path routing. Our approach decomposes the problem of finding an optimal unsplittable shortest path routing into the two subproblems of finding the optimal end-to-end routing paths and, afterwards, finding a routing metric that induces exactly these paths. The formulations we propose to compute the end-to-end routing paths do not

involve the routing lengths, but instead rely on the independence system characterization of unsplittable shortest path routings. This leads to mixed-integer linear programs that are smaller and stronger than those obtained with the traditional formulations involving also variables for the routing metric. The integer programming models, valid inequalities, and our implementation of this approach are described in Chapters 7 to 9.

1.3 Organization

This thesis is divided into three major parts. Part I is concerned with the combinatorial properties of those path sets that comprise unsplittable shortest path routings and with problems that are related to these path sets. In Part II, we study the computational complexity of basic network design and traffic engineering problems. In Part III, we finally develop an integer linear programming approach to solve network design and routing planning problems with unsplittable shortest path routings to optimality.

The two Chapters 2 and 3 precede these three parts. Chapter 2 serves as reference to the basic mathematical notions and notations used in this thesis. In Chapter 3, we describe the practical background, introduce the mathematical notions related to unsplittable shortest path routing, and formally define three basic planning problems that are considered throughout this thesis.

Part I is dedicated to the combinatorics of unsplittable shortest path routings and their compatible metrics. It comprises the two Chapters 4 and 5. Chapter 4 deals with the problem of finding a metric that induces a set of prescribed unique shortest paths or proving that no such metric exists. A problem version where the entire end-to-end paths are given and another version where only some arcs on these paths are given are considered. We review two linear programming formulations that can be used to solve these problems, provided the arc lengths of the metric are allowed to be fractional or arbitrarily large. We also show that the problems become computational hard if the arc lengths must be small integers, which is required in practice. Both the problem variant of finding integer arc lengths that minimize the longest arc length as well as the variant of finding integer arc lengths that minimize the longest path length are proven to be $\mathcal{APX}$-hard.

In Chapter 5, we study the combinatorial properties of unsplittable shortest path routings and discuss some related problems. We introduce an independence system which completely describes all those paths sets that correspond to an unsplittable shortest path routing. All previously known properties of these path sets, which are also reviewed (and generalized) in this section, are insufficient to characterize unsplittable shortest path routings. The independence system description cannot be represented by a finite list of forbidden path configurations of finite size (as most of the previously known

properties), but algorithmically it can be verified efficiently. We present a simple polynomial time algorithm that, given an arbitrary path set, either asserts that these paths form an unsplittable shortest path routing or finds an inclusion-wise minimal conflict among these paths. This result allows us to model and solve unsplittable shortest path routing problems the way we do in Part III of this thesis. The related optimization problems of finding an cardinality or weight minimal conflict in a given path set are both shown to be $\mathcal{NP}$-hard to approximate within a factor of $7/6 - \epsilon$. We also consider the opposite problem of finding a maximal set of paths that form an unsplittable shortest path routing. We present a polynomial time algorithm to find an inclusion-wise maximal such set in a given path set, and we show that the corresponding maximum cardinality and maximum weight versions are $\mathcal{NP}$-hard to approximate within a factor of $8/7 - \epsilon$. Analogous results are obtained for the arc-flow representation of unsplittable shortest path routings.

Part II of this thesis consists of Chapter 6 only. In this part, we discuss the relation between unsplittable shortest path routing and several other routing schemes and we study the computational complexity of the three basic unsplittable shortest path routing problems introduced in Chapter 3. We construct examples where the lowest possible link congestion that can be obtained with unsplittable shortest path routing exceeds the congestion achievable with multicommodity flow routing, shortest path routing with traffic splitting, or unsplittable flow routing by an arbitrarily large factor. We also show that the congestion minimization problem Min-Con-USPR is $\mathcal{NP}$-hard to approximate within a factor of $\mathcal{O}(|V|^{1-\epsilon})$, that the fixed charge network design problem FC-USPR is $\mathcal{NPO}$-complete, and that the capacitated network design problem Cap-USPR is inapproximable within a factor of $\mathcal{O}(2^{log^{1-\epsilon}|V|})$ in the directed and a factor of $2 - \epsilon$ in the undirected case. These results indicate that network design and routing optimization problems are indeed harder for unsplittable shortest path routing than for other routing schemes – both from the theoretical and from the practical point of view. In addition, we derive polynomial time approximation algorithms for various general and special cases of the considered problems.

In Part III of this thesis, we finally present a mixed–integer linear programming approach to solve network design and routing planning problems with unsplittable shortest path routing to optimality. This part consists of the three Chapters 7, 8, and 9. In Chapter 7, the basic mixed–integer linear programming models are introduced. In contrast to previous integer programming models for these (and similar) problems, our formulations contain no variables for the routing lengths. Instead, we introduce new inequalities to describe the valid routings in terms of arc or path routing variables only. For any feasible end-to-end routing computed with these models, a compatible routing metric can be easily computed in a post-processing step. We present two different formulation types – one based on path-flow variables

and the other one based on arc-flow variables, discuss the strength of their linear relaxations, and analyze the computational complexity of the respective separation and pricing problems. Several classes of valid inequalities for these models are discussed in Chapter 8.

In Chapter 9, we finally describe our implementation of the integer linear programming approach. Here we extend the basic mixed–integer linear programming models presented in Chapter 7 to the more realistic ones that have been used to solve the network design and traffic engineering problems for the DFN networks, describe the algorithm used to solve the problems, and finally report on the computational results obtained for the DFN networks G-WiN and X-WiN and for several benchmark instances.

Chapter 2

Mathematical Preliminaries

In the following, we review the basic definitions and concepts in linear algebra, graph theory, and computational complexity that are used throughout this thesis. This description does not serve as an introduction to these areas, it is meant only as a reference for the notions and notations used in the following chapters. We expect the reader to be familiar with the basic concepts treated here.

For an introduction into linear algebra, integer linear programming, and polyhedral combinatorics we recommend the books of Grötschel et al. [103], Nemhauser and Wolsey [146], and Schrijver [175]. The concepts in graph and hypergraph theory needed in this thesis are very basic and can be found in the textbooks of Berge [29] or Bondy and Murty [39], for example. For an introductory survey on independence systems and matroid theory see Welsh [189] or Bixby and Cunningham [32]. The basic concepts and notions in the field of computational complexity date back to Karp [121] and Garey and Johnson [96]. Papadimitriou [154] and Ausiello et al. [10] introduced the notions and complexity classes related to the approximability of problems, which are used throughout this thesis.

2.1 Linear Algebra

We denote the sets of real, rational, and integer numbers by $\mathbb{R}$, $\mathbb{Q}$, and $\mathbb{Z}$, respectively. For the non-negative real, rational, and integer numbers, we use the symbols $\mathbb{R}_+$, $\mathbb{Q}_+$, and $\mathbb{Z}_+$. The set of natural numbers without zero is denoted by $\mathbb{N}$. Given a real number $x \in \mathbb{R}_+$, $\lfloor x \rfloor$ denotes the largest integer number not larger than x and $\lceil x \rceil$ denotes the smallest integer number not smaller than x.

For a base set $\mathbb{K}$ and a finite index set E, $\mathbb{K}^E$ is the set of vectors consisting of $|E|$ components with values in $\mathbb{K}$. Each component of a vector $x \in \mathbb{K}^E$ is indexed by an element $e \in E$, i.e., $x = (x_e)_{e \in E}$. For $[n] := \{1, \ldots, n\}$ with $n \in \mathbb{N}$, we simply write $\mathbb{K}^n$ for $\mathbb{K}^{[n]}$. Given a set $F \subseteq E$, the vector $\chi^F \in \{0,1\}^E$ defined as $\chi^F_e = 1$ for all $e \in F$ and $\chi^F_e = 0$ for all $e \in E \setminus F$ is called

the **incidence vector** (or **characteristic vector**) of F. Conversely, the set $F_x := \{e \in E : x_e = 1\}$ is called the **incidence set** (or **characteristic set**) of a vector $x \in \{0,1\}^E$. More generally, we say that $S_x := \{e \in E : x_e \neq 0\}$ is the **support** of a vector $x \in \mathbb{R}^E$. The vectors of all 0's and of all 1's are denoted by $\mathbf{0} := \chi^{\emptyset}$ and $\mathbf{1} := \chi^E$, respectively.

Unless states otherwise, each vector is considered as a column vector and the superscript 'T' denotes the transposition of a vector. Addition of vectors, multiplication of vectors with scalars, and inner and outer products of vectors are defined as usual. For any finite set E, $\mathbb{R}^E$ and $\mathbb{Q}^E$ are vector spaces over the fields $\mathbb{R}$ and $\mathbb{Q}$, respectively. Given two vectors $x, y \in \mathbb{R}^E$, we write $x \leq y$ if $x_e \leq y_e$ for all $e \in E$, and $x \neq y$ if $x_e \neq y_e$ for some $e \in E$.

A vector $x \in \mathbb{R}^E$ is a **linear combination** of the vectors $x_1, x_2, \ldots, x_k \in \mathbb{R}^E$, if there exists some $\lambda \in \mathbb{R}^k$ with $x = \sum_{i=1}^k \lambda_i x_i$. If, in addition,

$$\left.\begin{array}{l} \lambda \geq 0 \\ \qquad \lambda^T \mathbf{1} = 1 \\ \lambda \geq 0, \quad \lambda^T \mathbf{1} = 1 \end{array}\right\} \text{ we call } x \text{ a } \left\{\begin{array}{c} \textbf{conic} \\ \textbf{affine} \\ \textbf{convex} \end{array}\right\} \textbf{ combination}$$

of the vectors $x_1, x_2, \ldots, x_k$. These combinations are **proper**, if $\lambda_i > 0$ for all $i = 1, \ldots, k$. Given a non-empty set $X \subseteq \mathbb{R}^E$, the symbols

$$\left.\begin{array}{c} lin(X) \\ cone(X) \\ aff(X) \\ conv(X) \end{array}\right\} \text{ denote the } \left\{\begin{array}{c} \textbf{linear} \\ \textbf{conic} \\ \textbf{affine} \\ \textbf{convex} \end{array}\right\} \textbf{ hull of the elements in } X.$$

We say that a set $X \subseteq \mathbb{R}^E$ is linearly or affinely **independent**, if none of its members is a proper linear or affine combination of the elements in X, respectively, otherwise X is called linearly or affinely **dependent**. The linear or affine **rank** of a set $X \subseteq \mathbb{R}^E$ is the maximum number of linearly or affinely independent vectors in X. The **dimension** $dim(X)$ of a set $X \subseteq \mathbb{R}^E$ is the affine rank of X minus 1. A set $X \subseteq \mathbb{R}^E$ with $dim(X) = |E|$ is called **full-dimensional**.

2.2 Linear and Integer Linear Programming

Any vector $a \in \mathbb{R}^n$, $a \neq \mathbf{0}$, and any scalar $\alpha \in \mathbb{R}$ together define a **linear inequality** $a^T x \leq \alpha$ with variables $x \in \mathbb{R}^n$. The set of all solutions $x \in \mathbb{R}^n$ to this inequality is the **half-space** $\{x \in \mathbb{R}^n : a^T x \leq \alpha\}$ in $\mathbb{R}^n$. The set of all solutions to the corresponding **linear equality** $a^T x = \alpha$ defines the **hyperplane** $\{x \in \mathbb{R}^n : a^T x = \alpha\}$.

A matrix $A \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$ define a **system of linear inequalities** $Ax \leq b$ for the variables $x \in \mathbb{R}^n$. Its solution set $P_{A,b} :=$

$\{x \in \mathbb{R}^n : Ax \leq b\}$ is called a **polyhedron**. Every polyhedron is the intersection of finitely many half-spaces. A polyhedron that is bounded (i.e., is contained in the convex hull of finitely many vectors) is called a **polytope**. A polyhedron that is also a cone is called a **polyhedral cone**.

An inequality $a^T x \leq \alpha$ is **valid** for a polyhedron P if $P \subseteq \{x \in \mathbb{R}^n : a^T x \leq \alpha\}$. For any valid inequality $a^T x \leq \alpha$, the set $F(P, a, \alpha) := \{x \in P : a^T x = \alpha\}$ is the **face** of P **defined** (or **induced**) by $a^T x \leq \alpha$. If $F(P, a, \alpha) \neq \emptyset$, then the inequality $a^T x \leq \alpha$ is called **tight** with respect to P. If $F(P, a, \alpha) = \{v\}$, then v is called a **vertex** of P. If $F(P, a, \alpha) \neq \emptyset$ and $dim(F(P, a, \alpha)) = dim(P) - 1$, then $F(P, a, \alpha)$ is a **facet** of P and $a^T x \leq \alpha$ is said to be a **facet-defining** inequality for P. The facets of a polyhedron are its inclusion-wise maximal faces. If P is full-dimensional, then the inequality defining a facet is unique up to scaling by a non-negative factor, i.e., if $a^T x \leq \alpha$ and $b^T x \leq \beta$ are both facet defining for P and $F(P, a, \alpha) = F(P, b, \beta)$, then $a = \lambda b$ and $\alpha = \lambda \beta$ for some $\lambda \in \mathbb{R}_+$.

Whether or not a system of linear inequalities has a solution can be characterized by the following lemma.

Lemma 2.1 (Farkas [87]) *A system of linear inequalities $Ax \leq b$ with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ has a solution $x \in \mathbb{R}^n$, if and only if there does not exist a vector $y \in \mathbb{R}_+$ with $y^T A = \mathbf{0}^T$ and $y^T b < 0$.*

Given a matrix $A \in \mathbb{R}^{m \times n}$, a vector $b \in \mathbb{R}^m$, and a vector $c \in \mathbb{R}^n$, the **linear programming problem** (in standard form) is to find a vector $x^* \in P_{A,b}$ that maximizes the linear function $c^T x$. This problem is written as

$$\max\{c^T x \ : \ Ax \leq b, \ x \in \mathbb{R}^n\} \ . \tag{P}$$

A vector $x \in \mathbb{R}^n$ satisfying $Ax \leq b$ is called a **feasible solution** of (P). A feasible solution x^* is an **optimal solution** of (P), if $c^T x^* \geq c^T x$ for all feasible solutions x of (P). The set of all optimal solutions of (P) is a face of the polyhedron $P_{A,b}$.

With every linear program (P) one can associate the so-called **dual linear program**

$$\min\{y^T b \ : \ y^T A = c^T, \ y \in \mathbb{R}^m_+\} \tag{D}$$

with variables $y \in \mathbb{R}^m_+$. The original linear program (P) is also called the **primal** program. The following fundamental theorem describes the connection between the primal and the dual linear program

Theorem 2.2 (Linear Programming Duality) *Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$, and consider the corresponding primal and dual linear programs* (P) *and* (D).