

# Routing and Capacity Optimization for IP Networks

vorgelegt von  
Dipl.-Math.oec. Andreas Bley  
Dessau

Von der Fakultät II — Mathematik und Naturwissenschaften  
der Technischen Universität Berlin  
zur Erlangung des akademischen Grades  
Doktor der Naturwissenschaften  
Dr. rer. nat.

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Volker Mehrmann

1. Bericht: Prof. Dr. Dr. h.c. Martin Grötschel

2. Bericht: Prof. Dr. Daniel Bienstock

Tag der Wissenschaftlichen Aussprache: 7. Februar 2007

Berlin 2007

D83



---

## Abstract

This thesis is concerned with dimensioning and routing optimization problems for communication networks that employ a shortest path routing protocol such as OSPF, IS-IS, or RIP. These protocols are widely used in the Internet. With these routing protocols, all end-to-end data streams are routed along shortest paths with respect to a metric of link lengths. The network administrator can configure the routing only by modifying this metric. In this thesis we consider the unsplittable shortest path routing variant, where each communication demand must be sent unsplit through the network. This requires that all shortest paths are uniquely determined.

The major difficulties in planning such networks are that the routing can be controlled only indirectly via the routing metric and that all routing paths depend on the same routing metric. This leads to rather complicated and subtle interdependencies among the paths that comprise a valid routing. In contrast to most other routing schemes, the paths for different communication demands cannot be configured independent of each other.

Part I of the thesis is dedicated to the relation between path sets and routing metrics and to the combinatorial properties of those path sets that comprise a valid unsplittable shortest path routing. Besides reviewing known approaches to find a compatible metric for a given path set (or to prove that none exists) and discussing some properties of valid path sets, we show that the problem of finding a compatible metric with integer lengths as small as possible and the problem of finding a smallest possible conflict in the given path set are both NP-hard to approximate within a constant factor.

In Part II of the thesis we discuss the relation between unsplittable shortest path routing and several other routing schemes and we analyze the computational complexity of three basic unsplittable shortest path routing problems. We show that the lowest congestion that can be obtained with unsplittable shortest path routing may significantly exceed that achievable with other routing paradigms and we prove several non-approximability results for unsplittable shortest path routing problems that are stronger than those for the corresponding unsplittable flow problems. In addition, we derive various polynomial time approximation algorithms for general and special cases of these problems.

In Part III of the thesis we finally develop an integer linear programming approach to solve these and more realistic unsplittable shortest path routing problems to optimality. We present alternative formulations for these problems, discuss their strength and computational complexity, and show how to derive strong valid inequalities. Eventually, we describe our implementation of this solution approach and report on the numerical results obtained for real-world problems that came up in the planning the German National Research and Education Networks G-WiN and X-WiN and for several benchmark instances.



## Zusammenfassung

Die Arbeit befasst sich mit der Kapazitäts- und Routenplanung für Kommunikationsnetze, die ein kürzeste-Wege Routingprotokoll verwenden. Diese Art von Protokollen ist im Internet weit verbreitet. Bei diesen Routingverfahren wird für jede Verbindung im Netz ein Längenwert festgelegt, diese Längen formen die sogenannte Routingmetrik. Die Routingwege der Kommunikationsbedarfe sind dann die jeweiligen kürzesten Wege bezüglich dieser Metrik. Bei der in der Arbeit untersuchten Variante dieser Routingprotokolle wird zusätzlich verlangt, dass es je Kommunikationsbedarf genau einen eindeutigen kürzesten Weg gibt.

Die Schwierigkeit bei der Planung solcher Netze besteht darin, dass sich die Routingwege einerseits nur indirekt über die Routingmetrik beeinflussen lassen, andererseits aber alle Routingwege von der gleichen Metrik abhängen. Dadurch können die Wege der verschiedenen Kommunikationsanforderungen nicht wie bei anderen Routingverfahren unabhängig voneinander gewählt werden.

Im ersten Teil der Arbeit werden der Zusammenhang zwischen gegebenen Wegesystemen und kompatiblen Routingmetriken sowie die Beziehungen der Wege eines zulässigen eindeutige-kürzeste-Wege-Routings untereinander untersucht. Dabei wird unter Anderem gezeigt, dass es NP-schwer ist, eine kompatible Metrik mit kleinstmöglichen Routinglängen zu einem gegebenen Wegesystem zu finden. Es wird auch bewiesen, dass das Finden eines kleinstmöglichen Konfliktes in einem gegebenen Wegesystem, zu dem keine kompatible Metrik existiert, NP-schwer ist.

Im zweiten Teil der Arbeit wird die Approximierbarkeit von drei grundlegenden Netz- und Routenplanungsproblemen mit eindeutige-kürzeste-Wege-Routing untersucht. Für diese Probleme werden stärkere Nichtapproximierbarkeitsresultate als für die entsprechenden Einwege-Routing Probleme bewiesen und es werden verschiedene polynomiale Approximationsverfahren für allgemeine und Spezialfälle entworfen. Ausserdem wird die Beziehung zwischen eindeutige-kürzeste-Wege-Routing und anderen Routingverfahren diskutiert.

Im dritten und letzten Teil der Arbeit wird ein (gemischt-) ganzzahliger Lösungsansatz für Planungsprobleme mit eindeutige-kürzeste-Wege-Routing vorgestellt. Für die im zweiten Teil diskutierten grundlegenden Netz- und Routenplanungsprobleme werden verschiedene (gemischt-) ganzzahlige lineare Modelle vorgestellt und es wird deren Lösbarkeit und die Stärke ihrer LP Relaxierungen untersucht. Es wird auch gezeigt, wie sich starke gültige Ungleichungen aus den in diesen Modellen enthalten Substrukturen ableiten lassen. Schließlich werden am Ende der Arbeit die Software-Implementierung dieses Lösungsverfahrens für eine praxisrelevante Verallgemeinerung der Planungsprobleme sowie die damit erzielten numerischen Ergebnisse vorgestellt und diskutiert.



## Acknowledgments

This work could not have been written without the help and encouragement of many people. I am extremely fortunate to be surrounded by a truly wonderful ensemble of fellow researchers and friends.

The greatest thanks goes to my supervisor Prof. Martin Grötschel for giving me the chance to work on this very interesting topic. His steady support and motivation gave me the freedom and the time I needed to finish my research.

My colleagues and the staff at the Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB) have provided a supportive and friendly working environment and I wish to express my gratitude to all of them. I am particularly grateful to Roland Wessäly, Adrian Zymolka, Arie M.C.A. Koster, Andreas Eisenblätter, Sebastian Orłowski, and Volker Kaibel for proof-reading several parts of the manuscript. Their feedback helped me to put things into the right perspective and made this thesis immeasurably better. Special thanks also go to the entire DISCNET team at ZIB and at Atesio GmbH for the effective cooperation in software development.

I would also like to thank my fellow researchers and friends at the Technical University Berlin, especially those in the Combinatorial Optimization and Graph Algorithms group, for lots of interesting mathematical seminars and discussions and many exciting basketball games.

This thesis was motivated by the problems arising in the planning and operation of the German national research and education network operated by the DFN-Verein. I have to thank all the people at the DFN-Verein and in particular Marcus Pattloch for an excellent cooperation in this research project. They ensured that the mathematics developed in line with the practical needs and they provided access to the real problems.

And last, but certainly not least, a very big thankyou to my family and all my friends for their understanding and encouragement throughout the course of this PhD.





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Focus of this Thesis . . . . .	1
1.2	Contributions of this Thesis . . . . .	3
1.3	Organization . . . . .	4
<b>2</b>	<b>Mathematical Preliminaries</b>	<b>7</b>
2.1	Linear Algebra . . . . .	7
2.2	Linear and Integer Linear Programming . . . . .	8
2.3	Graphs and Hypergraphs . . . . .	10
2.4	Walks, Paths, and Connectivity . . . . .	11
2.5	Independence systems and Matroids . . . . .	13
2.6	Computational Complexity and Approximation . . . . .	14
<b>3</b>	<b>Internet Routing and Planning Problems</b>	<b>21</b>
3.1	History of the Internet . . . . .	21
3.2	Architecture and Basic Functionality . . . . .	22
3.3	Shortest Path Routing . . . . .	24
3.4	Optimization Problems in IP networks . . . . .	28
3.5	Mathematical Model . . . . .	31
<b>I</b>	<b>Metrics and Routing Paths</b>	<b>37</b>
<b>4</b>	<b>The Inverse Unique Shortest Paths Problem</b>	<b>39</b>
4.1	Related Work . . . . .	40
4.2	Problem Definition . . . . .	41
4.3	Finding Real-Valued Lengths . . . . .	44
4.4	Inapproximability Results for Integer Lengths . . . . .	47
4.5	An LP-Rounding Algorithm . . . . .	56
4.6	Unique Shortest Path Forwardings . . . . .	60
4.7	Concluding Remarks . . . . .	66
<b>5</b>	<b>Unique Shortest Path Systems</b>	<b>69</b>
5.1	Related Work . . . . .	70

5.2	Basic Definitions and Observations . . . . .	71
5.3	Properties of Unique Shortest Path Systems . . . . .	75
5.4	Finding Irreducible Non-Unique Shortest Path Systems . . . . .	81
5.5	Finding Maximum Unique Shortest Path Systems . . . . .	90
5.6	Undirected Unique Shortest Paths Systems . . . . .	96
5.7	Unique Shortest Path Forwardings . . . . .	103
<b>II</b>	<b>Hardness and Approximability</b>	<b>111</b>
<b>6</b>	<b>Approximability of Unsplittable Shortest Path Routing</b>	<b>113</b>
6.1	Related Work . . . . .	115
6.2	Unsplittable Shortest Path Routing Problems . . . . .	117
6.3	Relation to Other Routing Schemes . . . . .	120
6.4	Inapproximability Results . . . . .	124
6.5	General Approximation Algorithms . . . . .	137
6.6	Special Cases . . . . .	144
6.7	Concluding Remarks . . . . .	153
<b>III</b>	<b>An Integer Programming Solution Approach</b>	<b>155</b>
<b>7</b>	<b>Integer Linear Programming Models</b>	<b>157</b>
7.1	Related Work . . . . .	158
7.2	Basics . . . . .	161
7.3	Path Routing Formulations . . . . .	164
7.4	Arc Routing Formulations . . . . .	167
7.5	Solving the LP Relaxations . . . . .	174
7.6	Strength of the LP Relaxations . . . . .	186
<b>8</b>	<b>Valid Inequalities</b>	<b>193</b>
8.1	Routing Inequalities . . . . .	194
8.2	Superadditive Metric Inequalities . . . . .	201
8.3	Precedence Constrained Knapsack Inequalities . . . . .	209
<b>9</b>	<b>Implementation and Computational Results</b>	<b>221</b>
9.1	Modeling the Real Problem . . . . .	222
9.2	Implementation . . . . .	237
9.3	Computational Results . . . . .	247
9.4	Conclusions and Future Work . . . . .	259
	<b>Bibliography</b>	<b>261</b>
	<b>Index</b>	<b>277</b>

# 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 approximate the minimum congestion that 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  $[n] := \{1, \dots, n\}$  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  $\chi_e^F = 1$  for all  $e \in F$  and  $\chi_e^F = 0$  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  $x_1, x_2, \dots, x_k \in \mathbb{R}^E$ , if there exists some  $\lambda \in \mathbb{R}^k$  with  $x = \sum_{i=1}^k \lambda_i x_i$ . If, in addition,

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

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

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

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

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

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**

$$\min\{y^T b : y^T A = c^T, y \in \mathbb{R}_+^m\} \quad (\text{D})$$

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).*

- (i) If both (P) and (D) have feasible solutions, then both (P) and (D) have optimal solutions and the optimal primal and the optimal dual objective function values are equal.
- (ii) If one of the programs (P) or (D) has no feasible solution, then the other is either unbounded or has no feasible solution.
- (iii) If one of the programs (P) or (D) is unbounded, then the other has no feasible solution.

If both (P) and (D) have feasible solutions, then the following theorem provides a useful characterization of optimal solutions.

**Theorem 2.3 (Complementary Slackness)** *Let  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ , and  $c \in \mathbb{R}^n$ , and suppose that the corresponding primal and dual linear programs (P) and (D) both have a feasible solution. Then  $x \in \mathbb{R}_+^n$  and  $y \in \mathbb{R}_+^m$  are optimal solutions of (P) and (D), respectively, if and only if  $y^T(b - Ax) = 0$  and  $x^T(y^T A - c) = 0$ .*

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 **integer linear programming problem** (in standard form) is to find an integer vector  $x^* \in \mathbb{Z}^n \cap P_{A,b}$  that maximizes the linear function  $c^T x$  over all vectors in  $\mathbb{Z}^n \cap P_{A,b}$ . This problem can be written as

$$\max\{c^T x : Ax \leq b, x \in \mathbb{Z}^n\}. \quad (\text{IP})$$

Relaxing the integrality restriction  $x \in \mathbb{Z}^n$  in (IP) to  $x \in \mathbb{R}^n$ , one obtains the **linear programming relaxation** (P) of (IP).

Applying several simple transformations, a linear or integer linear maximization problem can be turned into minimization problem, the underlying inequality system can be turned into an equality system (and vice versa), and a system with unbounded variables can be turned into a system with non-negative variables.

## 2.3 Graphs and Hypergraphs

A finite **graph** is a triple  $G = (V, E, \Psi)$  consisting of a non-empty finite set  $V$ , called the **nodes** (or vertices), a finite set  $E$ , called the **edges** (or links), and a relation of **incidence**  $\Psi : E \rightarrow V^{(2)}$  that associates with each edge two nodes, called its **ends** or **terminals**. Usually, we just write  $G = (V, E)$  and assume that the incidence relation is given implicitly in  $E$ .

Two nodes, which are the ends of an edge, are **adjacent** to one another (**neighbors**). A node that has no neighbors is **isolated**. The **degree**  $|\delta(v)|$  of a node  $v$  is the number of edges incident to  $v$ . An edge with identical ends is called a **loop**. If two edges join the same pair of ends, they are called **parallel**. A graph is **simple** if it has neither loops nor parallel edges.

A finite **digraph** (directed graph) is a triple  $D = (V, A, \Psi)$  consisting of a non-empty and finite set  $V$  called the **nodes**, a finite set  $A$  called the **arcs**, and a relation of **incidence**  $\Psi : A \rightarrow V^2$  that associates with each arc an ordered pair of nodes called its **ends** or **terminals**. Usually, we just write  $D = (V, A)$  and assume that the incidence relation is given implicitly in  $A$ .

For each arc  $a = (v, w)$ , we call  $v$  the **tail** and  $w$  the **head** of  $a$ . Parallel arcs and loops are defined as for graphs. Two arcs  $a = (v, w)$  and  $b = (w, v)$  are called **anti-parallel**. We call the graph  $G = (V, E)$  the **underlying graph** of the digraph  $D = (V, A)$  if there is a bijection between the arcs of  $D$  and the edges of  $G$ , such that for each arc  $a = (v, w) \in A$  there is an edge  $e = (v, w) \in E$  and for each edge  $e = (v, w) \in E$  the arc  $a = (v, w)$  or the arc  $a' = (w, v)$  is in  $A$ . The **associated digraph**  $D(G)$  of a graph  $G$  is the digraph obtained from  $G$  by replacing each edge by two antiparallel arcs with the same ends.

A finite **hypergraph** is a triple  $\mathcal{H} = (V, \mathcal{C}, \Psi)$  consisting of a non-empty finite set  $V$ , called the **nodes**, a finite set  $\mathcal{C}$ , called the **(hyper-)edges**, and a relation of **incidence**  $\Psi : \mathcal{C} \rightarrow 2^V$  that associates with each edge a subset of at least two nodes, called its **terminals**. Usually, we just write  $\mathcal{H} = (V, \mathcal{C})$  and assume that the incidence relation is given implicitly in  $\mathcal{C}$ .

A graph  $\hat{G} = (\hat{V}, \hat{E})$  is a **subgraph** of  $G = (V, E)$  if  $\hat{V} \subseteq V$  and  $\hat{E} \subseteq E$ . For a graph  $G = (V, E)$  and node set  $W \subseteq V$ , we denote by  $G - W$  the graph constructed by removing all nodes  $v \in W$  and all edges incident to at least one of these nodes from  $G$ . Analogously, for  $F \subseteq E$ , we denote by  $G - F$  the graph constructed by removing all edges  $e \in F$  from  $G$ . The subgraph **induced** by a node set  $W \subseteq V$  is  $G[W] := G - (V \setminus W)$ . The corresponding definitions and notions for directed graphs are analogous.

For a hypergraph  $\mathcal{H} = (V, \mathcal{C})$  and a node set  $W \subseteq V$ , the expression  $G - W$  denotes the hypergraph that is obtained by removing all nodes  $v \in W$  from the node set  $V$  and all edges that contain  $v$  and finally removing all edges that contain only one residual terminal. For  $F \subseteq \mathcal{C}$ , the sub-hypergraph  $G - F$  is obtained by removing all edges  $e \in F$  from  $\mathcal{H}$ . The sub-hypergraph **induced** by a node set  $W$  is  $G[W] := G - (V \setminus W)$ .

A subset  $S \subseteq V$  of the nodes of a graph  $G = (V, E)$  is called **stable** (or independent) if no edge  $e \in E$  is fully contained in  $S$ . In a hypergraph  $\mathcal{H} = (V, \mathcal{C})$ , a node set  $S \subseteq V$  is called **(weakly) stable** (or independent) if no hyperedge  $e \in \mathcal{C}$  is fully contained in  $S$ .

## 2.4 Walks, Paths, and Connectivity

A (directed) **walk** in the (directed) graph  $G$  is a sequence  $W = (v_0 e_1 v_1 \dots e_l v_l)$  of nodes  $v_0, \dots, v_l$  and edges (arcs)  $e_1, \dots, e_l$  of  $G$ , such that the nodes  $v_{i-1}$  and  $v_i$  are the ends of edge  $e_i$  (are head and tail of  $e_i$ , respectively) for each  $1 \leq i \leq l$ . We write  $e \in W$  or  $v \in W$  to indicate that some edge (or arc)  $e$

or some node  $v$  occur in the walk  $W$ . Vertex  $v_0$  is called the **source** of  $W$ , vertex  $v_l$  is called the **target** of  $W$ , and all other nodes are called **internal nodes** of  $W$ . We say that  $W$  is a **(directed)  $(v_0, v_l)$ -walk** or that  $W$  connects  $v_0$  and  $v_l$ . Given a walk  $W$ , we also refer to its source node by  $s_W$  and to its target node by  $t_W$ .

Given a vector  $c = (c_e)_{e \in E} \in \mathbb{R}^E$  (or  $c = (c_a)_{a \in A} \in \mathbb{R}^A$  in the directed case), the **length** of a walk  $W$  with respect to  $c$  is  $c(W) := \sum_{e \in W} c_e$ .

A (directed) walk  $W$  is called a **(directed) path** if all its nodes  $v_0, \dots, v_l$  are distinct. A (directed) walk  $W$  with  $v_0 = v_l$  is **closed**. A closed (directed) walk, whose edges (or arcs) are distinct, is a (directed) **circuit**. If the underlying (directed) graph is simple (i.e., contains neither loops nor parallel edges or arcs), then we refer to a path  $P = (v_0, a_1, v_1, \dots, a_l, v_l)$  with only its node sequence  $P = (v_0, v_1, \dots, v_l)$  for notational simplicity. Otherwise, we usually omit the nodes and write  $P = (e_1 \dots e_l)$ .

The **(u, v)-subpath** of a path  $P$  is denoted by  $P[u, v]$ . If  $P$  contains no  $(u, v)$ -subpath (or, in the directed case, the nodes  $u$  and  $v$  occur in reverse order in  $P$ ), we let  $P[u, v] := \emptyset$ . The **concatenation** of two paths  $P_1 = (v_0^1, a_1^1, \dots, v_l^1)$  and  $P_2 = (v_0^2, a_1^2, \dots, v_k^2)$  with  $v_l^1 = v_0^2$  is  $P_1 \oplus P_2 := (v_0^1, a_1^1, \dots, v_l^1 = v_0^2, a_1^2, \dots, v_k^2)$ .

Given a (directed) graph  $G$ , we denote the set of all paths in  $G$  by  $\mathcal{P}_G$ . The set of all  $(s, t)$ -paths between two nodes  $s, t \in V$  is denoted by  $\mathcal{P}_G(s, t)$ . For notational simplicity, we also write  $\mathcal{P}_G(K) := \bigcup_{(s, t) \in K} \mathcal{P}_G(s, t)$  for any set  $K \subseteq V \times V$ . If is clear from the context which (directed) graph  $G$  we refer to, the subscript  $G$  is typically omitted and we just write  $\mathcal{P}$ ,  $\mathcal{P}(s, t)$ , and  $\mathcal{P}(K)$ , respectively.

A collection of (directed)  $(s, t)$ -paths is called **node disjoint** (or internally disjoint) if no two paths have an internal node in common. We say a collection of (directed)  $(s, t)$ -paths is **edge disjoint** (or **arc disjoint**) if no two paths in the collection have an edge (an arc) in common.

An undirected graph  $G = (V, E)$  is a **forest** if it contains no circuit. If in addition  $G$  is connected, then it is called a **tree**. For each pair of nodes  $s, t \in V$ ,  $s \neq t$ , there is exactly one  $(s, t)$ -path in a tree. We say that an edge set  $F \subset E$  forms a forest or a **spanning tree** in  $G$  if the subgraph  $(G, F)$  is a forest or a tree, respectively.

A directed graph  $D = (V, A)$  is said to be an **arborescence** with root node  $r$  if its underlying undirected graph is a tree and there is a directed  $(r, t)$ -path in  $D$  for all  $t \in V \setminus \{r\}$ . If the underlying undirected graph is a tree and  $D$  contains a directed  $(s, t)$ -path for all nodes  $s \in V \setminus \{r\}$ , then  $D$  is an **anti-arborescence** with root  $r$ . Similar to the undirected case, we say that an arc set  $F \subset A$  defines a **spanning arborescence** or a **spanning anti-arborescence** with root  $r$  in  $D$  if the subgraph  $(D, F)$  is a spanning arborescence or anti-arborescence with root  $r$ , respectively.

An undirected graph  $G = (V, E)$  is said to be **connected** if there is a path between any two nodes. A graph  $G$  is **k-connected** if  $|V| > k$  and

for any  $W \subseteq V$  with  $|W| \leq k - 1$  the graph  $G - W$  is connected. This is equivalent to saying that there are at least  $k$  node disjoint paths between any pair of nodes of  $G$ . Analogously,  $G$  is called **k-edge-connected** if the deletion of any set of at most  $k - 1$  edges leaves a connected graph.

A directed graph  $D$  is said to be **strongly connected** if there is a directed path from any node to any other node.  $D$  is **strongly k-connected** if  $|V| > k$  and  $G - W$  is strongly connected for any  $W \subseteq V$  with  $|W| \leq k - 1$ .

The **components** of a graph  $G = (V, E)$  are the (inclusion) maximal connected subgraphs of  $G$ . The **strong components** of a digraph  $D = (V, A)$  are the maximal strongly connected subgraphs of  $D$ .

For any undirected graph  $G = (V, E)$  and any node set  $W \subseteq V$  the set of edges  $\delta(W) := \{(w, v) \in E : w \in W, v \in V \setminus W\}$  is a **cut**. If  $s \in W$  and  $t \notin W$  we call  $\delta(W)$  an **(s,t)-cut**. For a digraph  $D = (V, A)$  and a node set  $W \subseteq V$  we let  $\delta^+(W) := \{(w, v) \in A : w \in W, v \notin W\}$ ,  $\delta^-(W) := \delta^+(V \setminus W)$ , and  $\delta(W) := \delta^+(W) \cup \delta^-(W)$ . The arc-set  $\delta^+(W)$  is called a **directed cut**. For  $U, W \subseteq V$  we define  $[U, W] := \{(u, w) \in A : u \in U, w \in W\}$ .

Analogously, if  $G = (V, E)$  is a graph and  $U, S, W$  is a partition of  $V$  with  $U, W \neq \emptyset$  and there is no edge between  $U$  and  $W$ , we call  $S$  a **node cut** (or an articulation set). Equivalently,  $S$  is a node cut if and only if  $G - S$  is disconnected. If  $s \in U$  and  $t \in W$ , then we call  $S$  an **(s,t)-node cut** or say that  $S$  **separates**  $s$  and  $t$ .

## 2.5 Independence systems and Matroids

A (finite) **independence system** (or hereditary family) is a pair  $(S, \mathcal{I})$  of a finite ground set  $S$  and a collection of subsets  $\mathcal{I} \subseteq 2^S$  that satisfies the following conditions:

- (i)  $\emptyset \in \mathcal{I}$ , and
- (ii)  $I \in \mathcal{I}$  and  $J \subseteq I$  imply  $J \in \mathcal{I}$ .

The elements of  $\mathcal{I}$  are called **independent sets**. The subsets of  $S$  not in  $\mathcal{I}$  are called **dependent sets**. The **rank** of a subset  $J \subseteq S$  is defined as  $r(J) := \max\{|I| : I \subseteq J, I \in \mathcal{I}\}$ , i.e., the maximum cardinality of an independent set contained in  $J$ . The rank of the entire independence system  $(S, \mathcal{I})$  is  $r(S)$ .

An independence system can be characterized alternatively by

- (i) its independent sets  $\mathcal{I}$ ,
- (ii) its maximal independent sets, the so-called **bases**,
- (iii) its dependent sets,
- (iv) its minimal dependent sets  $\mathcal{C}$ , the so-called **circuits**,

(v) its rank function  $r : 2^S \rightarrow \mathbb{N}$ .

Here, maximal and minimal are meant with respect to set-inclusion. All these characterizations are equivalent, i.e., each description can be computed from any other. The equivalences between (i) and (ii) and between (iii) and (iv) are obvious. The remaining ones follow easily from the following three observations.

- (i)  $I \subseteq S$  is independent if and only if  $r(I) = |I|$ .
- (ii)  $I \subseteq S$  is independent if and only if it contains no circuit.
- (iii)  $C \subseteq S$  is dependent if and only if it is not contained in any base.

In many cases it is helpful to represent independence systems by hypergraphs. Given independence system  $(S, \mathcal{I})$ , its associated **circuit hypergraph** is  $\mathcal{H}(\mathcal{I}) = (S, \mathcal{C})$ , that is,  $\mathcal{H}(\mathcal{I})$  contains a node for every element  $i \in S$  and a hyperedge for every circuit  $C \in \mathcal{C}$ . The (weakly) stable sets  $I$  in the associated hypergraph  $\mathcal{H}(\mathcal{C})$  of an independence system  $(S, \mathcal{I})$  are exactly its independent sets  $I \in \mathcal{I}$ .

A well studied class of independence systems with nice algorithmic and structural properties are the so called matroids. A **matroid** is an independence system  $(S, \mathcal{I})$  that satisfies one of the following equivalent conditions:

- (i) For each  $I, J \in \mathcal{I}$  with  $|I| = |J| - 1$ , there is some  $j \in J \setminus I$  such that  $I \cup \{j\} \in \mathcal{I}$ .
- (ii) For each  $I, J \in \mathcal{I}$  with  $|I| < |J|$ , there is some  $K \subseteq J \setminus I$  such that  $|I \cup K| = |J|$  and  $I \cup K \in \mathcal{I}$ .
- (iii) For each  $I \subseteq S$ , all bases of  $I$  have the same size.

Matroids, as the name suggests, first appeared as an abstract generalization of matrices and much of the language in matroid and independence system theory is based on that of linear algebra. However, matroid and independence system theory proved to be a very useful unification of fundamental concepts of many different areas, e.g., lattice theory, graph theory, combinatorial optimization, partially ordered sets, and abstract simplicial complexes.

## 2.6 Computational Complexity and Approximation

In order to analyze the complexity of algorithms and problems, we need a formal description of problems, solutions, and algorithms.

Given a problem, a binary **encoding scheme** represents each **instance** and each **solution** as a string of symbols in the **alphabet**  $\Sigma = \{ '0', '1' \}$ .



With  $\Sigma^*$  we denote the set of all finite strings of symbols of the alphabet  $\Sigma$ . Formally, a **problem**  $\Pi$  (over an alphabet  $\Sigma$ ) is a pair  $\Pi = (\mathcal{I}, \mathcal{S})$  where

- $\mathcal{I} \subset \Sigma^*$  is the set of instances, and
- $\mathcal{S} : \mathcal{I} \rightarrow 2^{\Sigma^*}$  is a function that associates to any input instance  $I \in \mathcal{I}$  the set of feasible solutions.

The **encoding size**  $\langle I \rangle$  of a problem instance  $I$  w.r.t. an encoding scheme is the length of the string  $I$ . We assume that numbers are represented in the binary base and that the encoding scheme is reasonable, i.e., it provides a concise description of each instance and solution. In particular, the encoding size of a vector  $c \in \mathbb{Z}_+^k$  is not only its dimension  $k$  but the entire space required to encode its entries.

An algorithm is a program to solve a problem in some computer model. The most commonly used computer model is the Turing machine. In this model, an **algorithm**  $\text{ALG}$  is nothing but a finite deterministic  $m$ -tape Turing machine. We say that  $\text{ALG}$  **solves** the problem  $\Pi$  if it recognizes the admissible inputs  $I \in \mathcal{I}$  and computes feasible solutions in a finite number of steps, i.e., for each input string  $I \in \Sigma^*$  on the first tape and starting from a given beginning state  $B$ ,

- $\text{ALG}$  halts after a finite number of steps,
- if  $I \notin \mathcal{I}$ , then  $\text{ALG}$  halts in end state  $E_0$ , and
- if  $I \in \mathcal{I}$ , then  $\text{ALG}$  writes a string  $s \in \mathcal{S}(I)$  onto the second tape and halts in end state  $E_1$ .

With respect to polynomial time solvability of problems, this model is equivalent to other computer models, like the random access memory machine (RAM) model, for example. (However, it is usually more convenient to analyze an algorithm in the RAM model.)

The time required by an algorithm  $\text{ALG}$  on input  $I \in \Sigma^*$  is the number of elementary steps that  $\text{ALG}$  (i.e., the corresponding Turing machine) takes until it halts. If  $\text{ALG}$  does not halt, this time is  $\infty$ . The **time complexity** function  $T_{\text{ALG}} : \mathbb{Z}_+ \rightarrow \mathbb{Z}_+$  of an algorithm  $\text{ALG}$  is  $T_{\text{ALG}}(n) := \max\{\text{time of ALG on input } I : I \in \Sigma^* \text{ with } \langle I \rangle \leq n\}$ . If  $\text{ALG}$  halts on all (finite) inputs and there is a polynomial  $q$  with  $T_{\text{ALG}}(n) \leq q(n)$  for all  $n \in \mathbb{Z}_+$ , we call  $\text{ALG}$  a **polynomial algorithm**. Analogously, the **space complexity** function  $M_{\text{ALG}} : \mathbb{Z}_+ \rightarrow \mathbb{Z}_+$  of an algorithm  $\text{ALG}$  expresses the maximum space  $M_{\text{ALG}}(n)$  required by  $\text{ALG}$  on any input  $I \in \Sigma^*$  of size  $\langle I \rangle \leq n$ .

A **decision problem** is a problem that has either “yes” or “no” as solution. Each decision problem  $\Pi$  has an **associated language**

$$L(\Pi) = \{ I \in \mathcal{I} : I \text{ is a “yes” instance} \}.$$

A decision problem  $\Pi$  (or, more precisely, its associated language  $L(\Pi)$ ) belongs to the class  $\text{DTIME}(f)$  for some function  $f : \mathbb{Z}_+ \rightarrow \mathbb{Z}_+$ , if there exists an algorithm **ALG** with time complexity  $T_{\text{ALG}} \leq f$  that solves  $\Pi$  (i.e., decides  $L(\Pi)$ ). The problem  $\Pi$  belongs to the class  $\mathcal{P}$  (polynomially solvable) if there exists a polynomial time algorithm that solves  $\Pi$ .

A decision problem  $\Pi$  belongs to the class  $\mathcal{NP}$  (non-deterministic polynomially solvable), if there exists another decision problem  $\Pi' \in \mathcal{P}$  and a polynomial  $q$  such that, for each  $I \in \Sigma^*$ ,

$$I \in L(\Pi) \Leftrightarrow \exists s \in \Sigma^* \text{ with } \langle s \rangle \leq q(\langle I \rangle) \text{ such that } (I, s) \in L(\Pi') .$$

The string  $s$  is a polynomial size certificate if the answer to an instance  $I$  of  $\Pi$  is affirmative. (Often the certificate  $s$  is also called a solution for the problem instance  $I$ , although formally “yes” or “no” is the solution.) It is clear that  $\mathcal{P} \subseteq \mathcal{NP}$ , but whether  $\mathcal{P} = \mathcal{NP}$  or  $\mathcal{P} \neq \mathcal{NP}$  is still one of the major open questions in mathematics and computer science.

Let  $\Pi_1$  and  $\Pi_2$  be two decision problems (encoded in the same scheme on the same alphabet). Then  $\Pi_1$  is said to be **(Karp-)reducible** to  $\Pi_2$ , or short  $\Pi_1 \leq_K \Pi_2$ , if a function  $f : \Sigma^* \rightarrow \Sigma^*$  exist such that

- (i)  $I_1 \in L(\Pi_1)$  if and only if  $f(I_1) \in L(\Pi_2)$  and
- (ii)  $f$  is computable in polynomial time.

The function  $f$  is called a **(Karp-)reduction** (or a polynomial transformation).

A decision problem  $\Pi$  is called  **$\mathcal{NP}$ -hard** if  $\Pi' \leq_K \Pi$  for every problem  $\Pi' \in \mathcal{NP}$ . If  $\Pi$  is  $\mathcal{NP}$ -hard and  $\Pi \in \mathcal{NP}$ , then  $\Pi$  is called  **$\mathcal{NP}$ -complete**.

An **optimization problem** (over an alphabet  $\Sigma$ ) is a quadruple  $\Pi = (\mathcal{I}, \mathcal{S}, w, \text{goal})$ , where

- $\mathcal{I} \subset \Sigma^*$  is the set of instances,
- $\mathcal{S} : \mathcal{I} \rightarrow 2^{\Sigma^*}$  is a function that associates to any input instance  $I \in \mathcal{I}$  the set of feasible solutions  $\mathcal{S}(I)$ ,
- $w : \mathcal{I} \times \Sigma^* \rightarrow \mathbb{Q}_+$  is the objective function, defined only for pairs  $(I, s)$  with  $I \in \mathcal{I}$  and  $s \in \mathcal{S}(I)$ . For every such pair,  $w(I, s)$  provides a *strictly positive* number which is the value of the solution  $s$ .
- $\text{goal} \in \{\min, \max\}$  specifies whether  $\Pi$  is a minimization of a maximization problem.

Given an instance  $I \in \mathcal{I}$ , the set of optimal solutions is denoted by  $\mathcal{S}^*(I)$  and the optimal solution value by  $w^*(I)$ .

An optimization problem  $\Pi = (\mathcal{I}, \mathcal{S}, w, \text{goal})$  belongs to the class  $\mathcal{NPO}$ , if the following holds:

- (i) the set of instances  $\mathcal{I}$  is recognizable in polynomial time;
- (ii) there exists a polynomial  $q$  such that, given an instance  $I \in \mathcal{I}$ ,
  - $\langle s \rangle \leq q(\langle I \rangle)$  for each  $s \in \mathcal{S}(I)$ , and
  - for any  $s \in \Sigma^*$  with  $\langle s \rangle \leq q(\langle I \rangle)$ , it is decidable in polynomial time whether  $s \in \mathcal{S}(I)$ ;
- (iii) the objective function  $w$  is computable in polynomial time (for each pair  $(I, s)$  with  $I \in \mathcal{I}$  and  $s \in \mathcal{S}(I)$ ).

An  $\mathcal{NPO}$  problem is said to be **polynomially bounded**, if there is a polynomial  $q$  such that  $w(I, s) \leq q(\langle I \rangle)$  for all pairs  $(I, s)$  with  $I \in \mathcal{I}$  and  $s \in \mathcal{S}(I)$ . The class  $\mathcal{NPO-PB}$  is the set of all polynomially bounded  $\mathcal{NPO}$  problems.

An optimization problem belongs to the class  $\mathcal{PO}$  if it is in  $\mathcal{NPO}$  and there exists a polynomial time algorithm **ALG** that, for any instance  $I \in \mathcal{I}$ , returns an optimal solution  $s \in \mathcal{S}^*(I)$ . Optimization problems in  $\mathcal{PO}$  are polynomially solvable.

For each optimization problem  $\Pi = (\mathcal{I}, \mathcal{S}, w, goal)$ , we can define its **associated decision problem**  $\Pi_D$  by fixing a target objective value  $\bar{w} \in \mathbb{Q}_+$  as follows: “Is there a feasible solution  $s \in \mathcal{S}(I)$  with  $w(I, s) \geq \bar{w}$ ?” if  $goal = \max$  or “Is there a feasible solution  $s \in \mathcal{S}(I)$  with  $w(I, s) \leq \bar{w}$ ?” if  $goal = \min$ . The optimization problem  $\Pi$  is called  **$\mathcal{NP}$ -hard** if its associated decision problem  $\Pi_D$  is  $\mathcal{NP}$ -hard.

Clearly, if an  $\mathcal{NPO}$  problem can be solved in polynomial time, then its corresponding decision problem can also be solved in polynomial time. As a consequence, if  $\mathcal{P} \neq \mathcal{NP}$ , then any  $\mathcal{NPO}$  problem whose corresponding decision problem is  $\mathcal{NP}$ -complete is not solvable in polynomial time. In these cases we sacrifice optimality and start looking for approximate solutions computable in polynomial time.

Given an instance  $I$  and a feasible solution  $s \in \mathcal{S}(I)$ , we define the **performance ratio** (or approximation ratio) of  $s$  with respect to  $I$  as

$$R(I, s) = \max \left\{ \frac{w(I, s)}{w^*(I)}, \frac{w^*(I)}{w(I, s)} \right\} .$$

The performance ratio is always a number greater than or equal to 1 and is as close to 1 as  $s$  is close to the optimum solution.

Let  $\Pi$  be an  $\mathcal{NPO}$  problem and let **ALG** be an algorithm that, for any instance  $I$  of  $\Pi$  with  $\mathcal{S}(I) \neq \emptyset$ , returns a feasible solution  $\text{ALG}(I) \in \mathcal{S}(I)$ . Given an arbitrary function  $r : \mathbb{Z}_+ \rightarrow (1, \infty)$ , we say that **ALG** is an  **$r(n)$ -approximation algorithm** for  $\Pi$  if

$$R(I, \text{ALG}(I)) \leq r(\langle I \rangle) ,$$

for any instance  $I \in \mathcal{I}$ . If  $\Pi$  admits a *polynomial time*  $r(n)$ -approximation algorithm we say that it is **approximable within (a factor of)  $r(n)$** . Given a class of functions  $F$ ,  $F$ - $\mathcal{APX}$  is the class of all  $\mathcal{NPO}$  problems  $\Pi$  that admit a polynomial time  $r(n)$ -approximate algorithm for  $\Pi$  for some  $r \in F$ . In particular,  $\mathcal{APX}$ ,  $\log$ - $\mathcal{APX}$ ,  $\text{poly}$ - $\mathcal{APX}$ , and  $\text{exp}$ - $\mathcal{APX}$  denote the classes  $F$ - $\mathcal{APX}$  with  $F$  equal to the set of functions  $\mathcal{O}(1)$ , the set  $\mathcal{O}(\log(n))$ , the set  $\cup_{k>0} \mathcal{O}(n^k)$ , and the set  $\cup_{k>0} \mathcal{O}(2^{n^k})$ , respectively.

An algorithm  $\text{ALG}$  is said to be an **approximation scheme** for  $\Pi$  if, for any instance  $I$  of  $\Pi$  and for any rational  $r > 1$ ,  $\text{ALG}$  returns a feasible solution  $\text{ALG}(I, r)$  of  $I$  whose performance ratio is at most  $r$ . An  $\mathcal{NPO}$  problem  $\Pi$  belongs to the class  $\mathcal{PTAS}$  if it admits a polynomial time approximation scheme, that is, an approximation scheme whose time complexity is bounded by  $q(\langle I \rangle)$  where  $q$  is a polynomial. Observe that the time complexity of an approximation scheme may be exponential in  $r$ . Thus, computations with values  $r$  very close to 1 may turn out to be practically infeasible. This leads to the notion of a fully polynomial time approximation scheme.

An  $\mathcal{NPO}$  problem  $\Pi$  belongs to the class  $\mathcal{FPTAS}$  if it admits a **fully polynomial time approximation scheme**, that is, an approximation scheme whose time complexity is bounded polynomially in both  $\langle I \rangle$  and  $\frac{1}{r-1}$ . Clearly, the following inclusions hold:

$$\begin{aligned} \mathcal{PO} \subseteq \mathcal{FPTAS} \subseteq \mathcal{PTAS} \subseteq \mathcal{APX} \subseteq \log\text{-}\mathcal{APX} \subseteq \text{poly}\text{-}\mathcal{APX} \\ \subseteq \text{exp}\text{-}\mathcal{APX} \subseteq \mathcal{NPO} . \end{aligned}$$

It is also easy to see that these inclusions are strict if and only if  $\mathcal{P} \neq \mathcal{NP}$ .

Similar to the decision problems, there is a notion of completeness for optimization problems. This notion is based on approximation preserving based reductions. Let  $\Pi_1 = (\mathcal{I}_1, \mathcal{S}_1, w_1, \text{goal}_1)$  and  $\Pi_2 = (\mathcal{I}_2, \mathcal{S}_2, w_2, \text{goal}_2)$  be two optimization problems in  $\mathcal{NPO}$ . Then  $\Pi_1$  is said to be **AP-reducible** to  $\Pi_2$ , or short  $\Pi_1 \leq_{AP} \Pi_2$ , if two functions  $f$  and  $g$  and a positive constant  $\alpha \geq 1$  exist such that:

- (i) for any instance  $I_1 \in \mathcal{I}_1$  and for any rational  $r > 1$ ,  $f(I_1, r) \in \mathcal{I}_2$ ,
- (ii) for any instance  $I_1 \in \mathcal{I}_1$  and for any rational  $r > 1$  with  $\mathcal{S}_1(I_1) \neq \emptyset$ ,  $\mathcal{S}_2(f(I_1, r)) \neq \emptyset$ ,
- (iii) for any instance  $I_1 \in \mathcal{I}_1$ , for any rational  $r > 1$ , and for any  $s_2 \in \mathcal{S}_2(f(I_1, r))$ ,  $g(I_1, s_2, r) \in \mathcal{S}_1(I_1)$ ,
- (iv) both functions  $f$  and  $g$  are computable in polynomial time for any fixed rational  $r > 1$ , and
- (v) for any instance  $I_1 \in \mathcal{I}_1$ , for any rational  $r > 1$ , and for any  $s_2 \in \mathcal{S}_2(f(I_1, r))$ ,

$$R(f(I_1, r), s_2) \leq r \text{ implies } R(I_1, g(I_1, s_2, r)) \leq 1 + \alpha(r - 1) .$$

$$\begin{array}{ccc}
I_1 \in \mathcal{I}_1 & \xrightarrow{\text{ALG}_1} & g(I_1, s_2, r) \in \mathcal{S}_1 \\
f \downarrow & & \uparrow g \\
f(I_1, r) \in \mathcal{I}_2 & \xrightarrow{\text{ALG}_2} & s_2 \in \mathcal{S}_2
\end{array}$$

**Figure 2.1** Reducibility between problems

The triple  $(f, g, \alpha)$  is called an **AP-reduction** (approximation preserving reduction) from  $\Pi_1$  to  $\Pi_2$ . If  $\Pi_1 \leq_{AP} \Pi_2$ , then any  $r(n)$ -approximate algorithm  $\text{ALG}_2$  for  $\Pi_2$  carries over to an  $(1 - \alpha(r(n) - 1))$ -approximate algorithm  $\text{ALG}_1$  for  $\Pi_1$ , see Figure 2.1. Hence, if  $\Pi_1 \leq_{AP} \Pi_2$  and  $\Pi_2 \in \mathcal{APX}$  (respectively  $\Pi_2 \in \mathcal{PTAS}$ ), then  $\Pi_1 \in \mathcal{APX}$  (respectively  $\Pi_1 \in \mathcal{PTAS}$ ).

Another type of approximation preserving reducibility is the so called L-reducibility introduced by Papadimitriou and Yannakakis [155, 156]. This notion is based on showing both a linear relation between the optimal objective values and a linear relation between the absolute errors of corresponding solutions. An  $\mathcal{NPO}$  problem  $\Pi_1$  is said to be **L-reducible** to another  $\mathcal{NPO}$  problem  $\Pi_2$ , in symbols  $\Pi_1 \leq_L \Pi_2$ , if two functions  $f$  and  $g$  and two positive constants  $\beta$  and  $\gamma$  exist such that:

- (i) for any instance  $I_1 \in \mathcal{I}_1$ ,  $f(I_1, r) \in \mathcal{I}_2$ ,
- (ii) for any instance  $I_1 \in \mathcal{I}_1$  with  $\mathcal{S}_1(I_1) \neq \emptyset$ ,  $\mathcal{S}_2(f(I_1)) \neq \emptyset$ ,
- (iii) for any instance  $I_1 \in \mathcal{I}_1$  and for any  $s_2 \in \mathcal{S}_2(f(I_1))$ ,  $g(I_1, s_2) \in \mathcal{S}_1(I_1)$ ,
- (iv) both functions  $f$  and  $g$  are computable in polynomial time,
- (v) for any instance  $I_1 \in \mathcal{I}_1$ ,

$$w_2^*(f(I_1)) \leq \beta \cdot w_1^*(I_1) , \text{ and}$$

- (vi) for any instance  $I_1 \in \mathcal{I}_1$  and for any  $s_2 \in \mathcal{S}_2(f(I_1))$ ,

$$|w_1^*(I_1) - w_1(I_1, g(I_1, s_2))| \leq \gamma \cdot |w_2^*(f(I_1)) - w_2(f(I_1), s_2)| .$$

The quadruple  $(f, g, \beta, \gamma)$  is called an **L-reduction** (linear reduction) from  $\Pi_1$  to  $\Pi_2$ . Although the notion of L-reducibility is not as powerful as that of AP-reducibility, it is very useful. Within  $\mathcal{APX}$ , the existence of an L-reduction implies the existence of an AP-reduction. If  $\Pi_1 \leq_L \Pi_2$  and  $\Pi_2 \in \mathcal{APX}$ , then any  $r(n)$ -approximate algorithm  $\text{ALG}_2$  for  $\Pi_2$  carries over to an  $1 + \beta\gamma(1 - r(n))$ -approximate algorithm  $\text{ALG}_1$  for  $\Pi_1$ , i.e., also  $\Pi_1 \in \mathcal{APX}$ .

Given a class  $\mathcal{C}$  of  $\mathcal{NPO}$  problems, a problem  $\Pi$  is  **$\mathcal{C}$ -hard** (with respect to AP-reducibility) if  $\Pi' \leq_{AP} \Pi$  for any  $\Pi' \in \mathcal{C}$ . A  $\mathcal{C}$ -hard problem  $\mathcal{C}$ -complete (with respect to AP-reducibility) if it belongs to  $\mathcal{C}$ .

It can be shown that, unless  $\mathcal{P} = \mathcal{NP}$ , if an  $\mathcal{NPO}$  problem  $\Pi$  is  $\mathcal{APX}$ -hard, then it does not belong to  $\mathcal{PTAS}$ , i.e., there is a threshold value  $\epsilon > 0$  such that  $\Pi$  cannot be approximated within a factor of  $1 + \epsilon$ . Similar, if  $\Pi$  is  $\mathcal{NPO}$ -complete, then it does not belong to  $\mathcal{APX}$  and cannot be approximated within a factor of  $2^{n^\epsilon}$ , for some  $\epsilon > 0$ .

## Chapter 3

# Internet Routing and Planning Problems

In recent years, the Internet has evolved to the most dominant world-wide communication network. The main design principles that made it such a success are robustness, simplicity, and decentralization. Internet technology was designed to “do its best” to handle the data traffic, not to guaranty a particular level of end-to-end performance or Quality-of-Service. Especially the rather simple and decentralized routing mechanisms with automatic re-configuration in case of topology changes or network failures made it possible to interconnect and operate many heterogeneous networks with a minimum of administrative overhead.

The aim of this chapter is to introduce the reader to the main functional aspects of networks that are based on Internet Protocol (IP) routing (or other shortest-path routing protocols) and to illustrate the problems that typically arise in planning and operating such networks. We will avoid daunting specifications and technicalities in our description as far as possible. A more detailed description of the protocols and networking concepts can be found in the books of Perlman [159] or Stevens [181].

We begin with a brief glimpse into the history of the Internet in Section 3.1. After describing the general architecture of the Internet in Section 3.2, we discuss the shortest path routing mechanisms that are employed within its autonomous subnetworks in Section 3.3. In Section 3.4, we describe the main planning tasks that arise in the design and operation of such a network. Finally, in Section 3.5, we present a mathematical description of these problems.

### 3.1 History of the Internet

The history of the Internet dates back to 1962, when the *RAND Corporation* started research on robust distributed computer networks. Two years later, the *Advanced Research Project Agency* ARPA began sponsoring re-

search on “interconnecting time-sharing computers”, and in 1969 the network ARPANET connecting Stanford University, UCLA, UCSB, and University of Utah was created. In 1971, ARPANET connected 23 sites all over the United States. The Royal College in the United Kingdom was the first European institution that joined the network in 1973. The first network discussion board USENET, a predecessor of today’s World-Wide-Web, was invented in 1973.

The years 1983 and 1984 can be regarded as the birth of the Internet. In these years, the TCP/IP routing mechanisms and the Domain Name System (DNS) were introduced. These new automated routing protocols together with the name-based addressing of the hosts admitted the rapid growth of the network. By 1987, already more than 10,000 hosts were connected to the Internet.

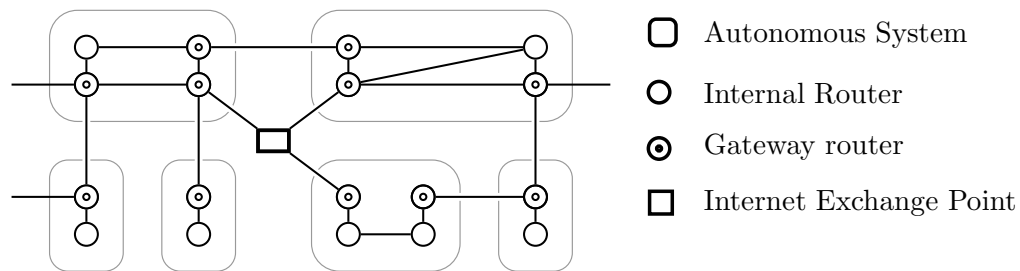
After 1990, when the Hyper-Text Markup Language (HTML) and the World-Wide-Web (WWW) were established, the large scale commercialization of the Internet started. The number of connected hosts and available services increased rapidly. Over the last decade, the traffic volume transmitted via the Internet approximately doubled every year [60]. In the 1995–1996 boom, when the first graphic-intensive web browsers emerged, the Internet’s traffic volume even doubled every three months.

Nowadays, the Internet is the most dominant communication network, connecting about 400 million hosts (in January 2006) and providing a uniform infrastructure for diverse applications that previously have been served by dedicated networks.

## 3.2 Architecture and Basic Functionality

The Internet is a huge, heterogeneous, virtual network. It is comprised of more than 13,000 distinct networks, so-called **Autonomous Systems (AS)** or **routing domains**. Each AS is an independently managed network consisting of its own routers and links. Traditionally, each AS was an individual provider’s network, such as a national research and educational network, a company’s private network, or the network of some public Internet service provider, but nowadays many large Internet service providers operate more than one AS. Connectivity between different ASes is provided by so-called peering (or uplink) connections. In each AS, there are one or more dedicated gateway (or border) routers that are linked either directly to gateway routers in other ASes or to public Internet exchange points, as illustrated in Figure 3.1. Since the decommission of the *National Science Foundation Network (NSFNet)*, which was the national scientific research and educational network in the USA, in 1995 the Internet has no central core network. All ASes in the Internet use the **Internet Protocol (IP)** [166, 167] for data transmission. The different application data streams to





**Figure 3.1** Global architecture of the Internet.

be transmitted are broken up into standardized IP **packets** (or datagrams) at their origin, sent through the network packet by packet, and reassembled to the original data streams at their respective destinations. Each IP packet consists of a header and the body (or payload). The header contains the source and the destination addresses, the length of the packet, and several other attribute fields. The packet body contains the application data. The splitting-up of the application data and the reassembling of the packets is done by a transport layer protocol, typically the Transport Control Protocol (TCP) [168] or the User Datagram Protocol (UDP) [165]. These protocols also are responsible for all quality and reliability issues of the transmission, such as controlling the transmission rate or requesting the re-transmission of lost packets. The IP network layer is only responsible for routing the individual packets through the network “at its best”, meaning that there is no guarantee that all packets indeed reach their destination or even arrive within a given time or in the same order they were sent.

According to its decentralized architecture, the Internet has a two-level routing mechanism. The **inter-domain routing** across several ASes is controlled by an Exterior Gateway Protocol (EGP). Currently, the **Border Gateway Protocol (BGP)** [171, 170] is the only used and de-facto standard EGP in the Internet.<sup>1)</sup> BGP is a policy based protocol. Peering gateway routers exchange information on the paths to the different ASes and, based on the paths reported by its neighbors, every gateway router then selects its preferred route to each destination. Which routes are announced to or accepted from peering routers is controlled by the router’s list of policies. Every BGP router can filter and modify the information it announces to its neighbors. This allows Internet service providers to configure the inter-AS routing that affects their AS according to national and international laws, their own security and safety policies, and the transit or service agreements they made with other providers. BGP announces only the AS-path to a destination, that is, only the identifiers of the different ASes along the path. The internal topology of an AS is not visible to the routers in other ASes. By default, a gateway router chooses the shortest AS-path that fulfills all policy rules as the preferred route to a destination.

<sup>1)</sup>The names *Autonomous System* and *Gateway router* are BGP terminology, in fact.

It is also important to note that BGP only standardizes the data formats to automatically announce routes and network failures. It neither defines a minimal set of information a gateway router must transmit to its peers nor does it specify how the inter-domain routes are computed from this information. In particular, BGP cannot guarantee that the network converges to a stable inter-AS routing, see Griffin and Wilfong [100].

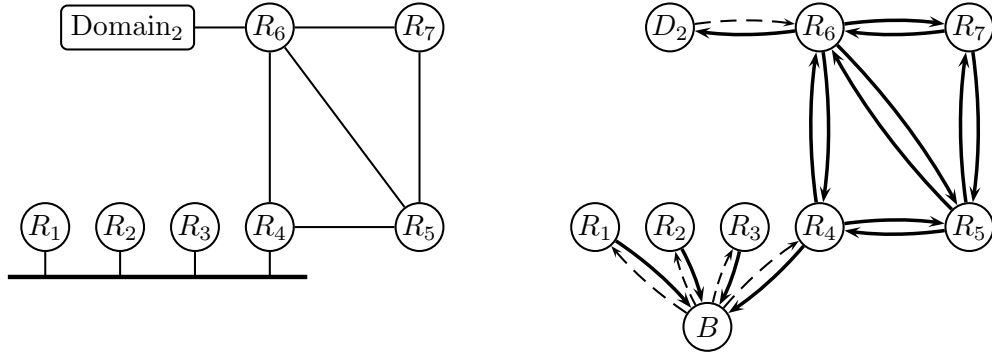
The **intra-domain routing** within a single AS is controlled by so-called Interior Gateway Protocols (IGP). In principle, each network provider can use her favorite IGP and can configure the routing within her own AS as she likes. The most commonly used IGPs today are *Open Shortest Path First* (OSPF) and *Intermediate System to Intermediate System* (IS-IS). These protocols are so-called shortest path routing protocols, which send each IP packet along a shortest path from its source to its destination. In the following section, we describe their routing mechanisms in more detail.

### 3.3 Shortest Path Routing

The basic functionality of all shortest path routing protocols is the same: Given a **metric** of administrative link lengths, they compute shortest end-to-end paths with respect to this metric and send each packet along a shortest path to its destination. The various protocols differ mainly in the way this functionality is realized.

The **Routing Information Protocol (RIP)** [114] used in the very first version of the Internet is a so-called **distance vector protocol**, where neighboring routers regularly exchange information about the distance to all known destinations. The shortest path computation is performed by all routers together with a distributed Bellman-Ford algorithm [22, 91]. RIP is a very restricted shortest path routing protocol. It always uses unit-lengths for the link metric, no other metric can be supplied. Furthermore, RIP admits only distances between 1 and 15 with respect to this unit-length metric, so routers that are 16 or more hops apart seem to be disconnected. In the small ARPANET network of the 1970s these restrictions were tolerable, but when the Internet started to grow in the 1980s this routing protocol needed to be replaced.

The protocols **Open Shortest Path First (OSPF)** [143] and **Intermediate System to Intermediate System (IS-IS)** [54] used nowadays are **link state protocols**. Here, each router periodically advertises the state of its adjacent links to all other routers in the AS. In a converged state, each router has full knowledge about the topology of the AS and computes its shortest paths to all other destinations locally using Dijkstra's algorithm [73]. Both protocols OSPF and IS-IS use a directed model of the underlying network. Figure 3.2 illustrates the graph model used in OSPF. By assigning a routing length (or routing weight) to each outgoing link interface of



(a) Example AS. The routers  $R_1$  to  $R_4$  are connected by a broadcast link, all other connections are bidirectional point-to-point links. Router  $R_6$  is connected by a bidirectional point-to-point link to another AS.

(b) Directed OSPF graph model. The routing lengths of the solid arcs can be set by the administrator at the respective router-to-link interfaces in the network. The dashed arcs represent link-to-router interfaces, whose routing lengths are fixed to 0.

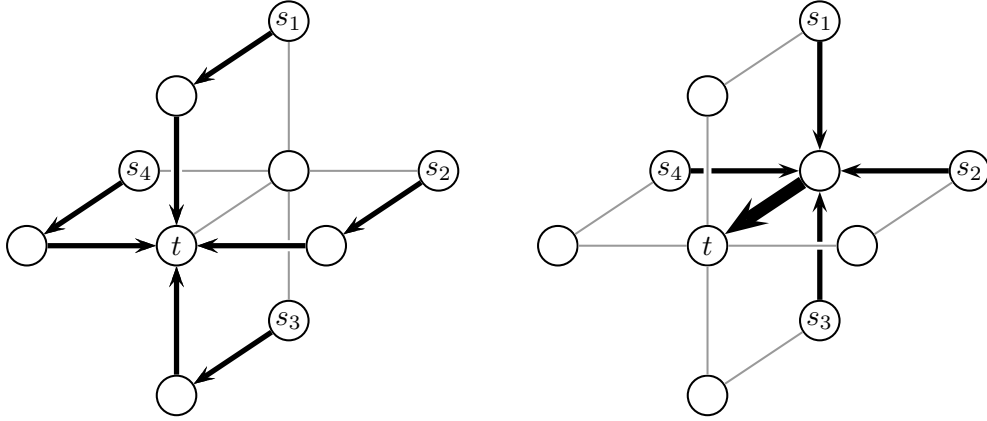
**Figure 3.2** Example AS and its directed OSPF graph model. Point-to-point links between two routers are represented by pairs of anti-parallel arcs between the corresponding router vertices. Broadcast link is modeled by an artificial vertex and all routers are connected to this vertex by a pair of anti-parallel arcs.

each router, the network operator can supply an individual routing metric. OSPF admits arbitrary routing lengths between 1 and  $2^{16} - 1 = 65535$  for the link interfaces, while the original version of IS-IS admits only lengths between 1 and 63. In principle, both protocols admit asymmetric routing metrics with different lengths for the different interfaces (i.e., directions) of a link. In order to simplify the network management, however, several network providers allow only symmetric routing metrics in their ASes.

Traditionally, IP networks operate in a single path routing mode without traffic splitting, which means that all packets from one origin to one destination follow the same path. Since neither OSPF nor IS-IS explicitly specify how to resolve ambiguities if multiple shortest paths exist, the routing metric should be chosen such that all shortest paths are unique. Otherwise, it would depend on the actual implementation of the routing protocol which of the shortest paths is used and the network provider would lack control over the routing, as illustrated in Figure 3.3.

OSPF also supports a **multi-path** variant of shortest path routing called **equal cost multi-path (ECMP)** routing. If there are multiple shortest paths towards a particular destination at the current router, then the traffic towards this destination is split (almost) equally among the outgoing links contained in these paths.<sup>2)</sup> Figure 3.4 illustrates such a routing. Compared

<sup>2)</sup>Perfectly equal traffic splitting is not achievable in practice. If packets with the same destination were distributed equally among the corresponding outgoing links in a packet-by-packet round robin manner, packets belonging to the same application data stream might be sent via different paths and arrive at the destination in reverse order. To avoid



(a) Routing with balanced traffic flows. (b) Routing with unbalanced traffic flows.

**Figure 3.3** Example for an ambiguous shortest path routing. Suppose there is a traffic stream from each of the four routers  $s_i$ ,  $i = 1, \dots, 4$ , to router  $t$ . If all routing lengths are 1, then it is not clear along which paths the corresponding packets are routed. Both the left or the right routing might be chosen by the routers.

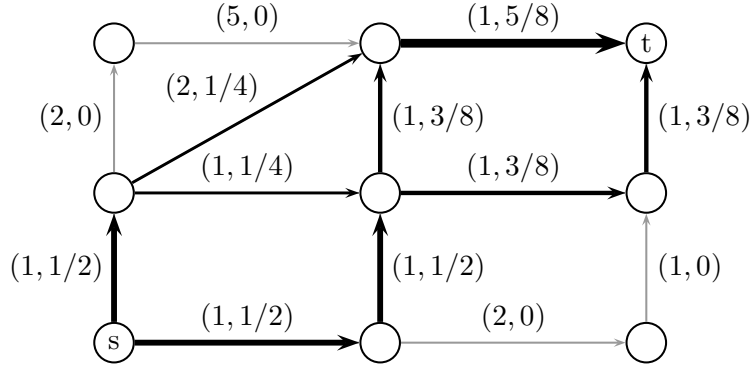
to the traditional single shortest path routing variant, ECMP routing provides enhanced possibilities to balance the traffic flows within an AS. On the other hand, however, it also complicates network monitoring and management. Therefore, some network providers insist on using traditional single shortest path routing without traffic splitting in their ASes.

The two main advantages of shortest path routing are that its decentralized concept requires no central control intelligence and that the real-time packet forwarding operations can be implemented very efficiently. In practice, every router maintains a local **forwarding table**, which maps the possible destination addresses to its outgoing links. Because each packet is sent along a shortest path towards its destination, all packets with the same destination address leave the router via the same link(s),<sup>3)</sup> independent of their origin.<sup>4)</sup> In order to decide to which of its neighbors to forward a packet, a router only needs to inspect the destination address in the packet header and look-up the right outgoing link in its forwarding table. The routing protocols OSPF or IS-IS run as asynchronous processes that co-

such packet disordering problems, most router vendors implement ECMP such that either all packets sent to the same host or all packets belonging to the same communication session follow the same path, which leads to non-uniform traffic splitting. Furthermore, IP packets have variable size. So even a packet-by-packet round robin approach would yield a non-uniform split.

<sup>3)</sup>For ECMP routing, the forwarding table may contain several outgoing links per destination address.

<sup>4)</sup>Recent protocols, such as *Multi-Protocol Label Switching* (MPLS) [173], as well as traditional connection-based protocols, like *Private Network to Network Interface* (PNNI) [164], allow routing and forwarding decisions to depend on both the source and the destination addresses and further header fields.



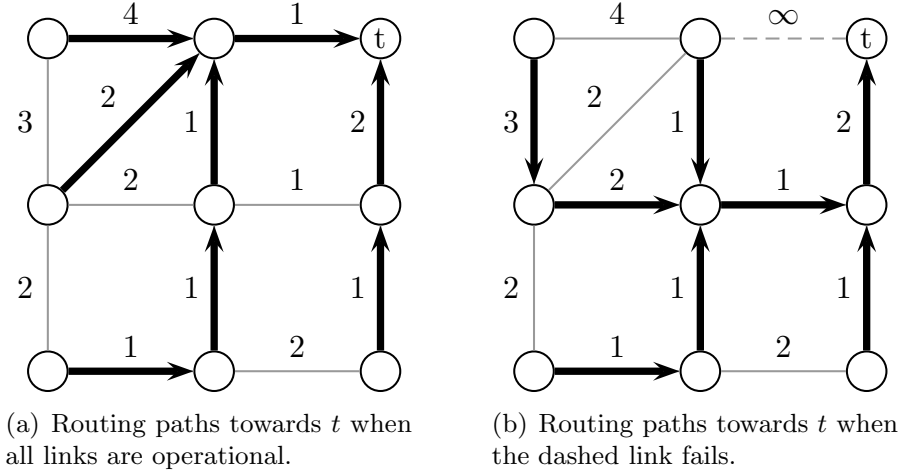
**Figure 3.4** ECMP routing example. The first value at each arc is the routing length, the second value is the fraction of the  $(s, t)$ -flow across this arc.

ordinate the forwarding tables of the routers in the AS. As the forwarding table sizes grow only linearly with the number of possible destination addresses, the table lookup operation requires very little computational effort per packet even in huge network.<sup>5)</sup> Modern high-speed routers have the packet forwarding functionalities implemented in specialized hardware and achieve enormous throughput rates.

Another important advantage of shortest path routing is its inherent robustness against network failures. The routers exchange network state information and recompute the shortest paths in regular time intervals and upon critical events. If a router detects that one of its adjacent links (or neighboring routers) fails or becomes functional again, it propagates this information to all other routers in the AS. Each router then updates its network topology database, recomputes its shortest paths in the residual network, and updates the forwarding tables accordingly. This way, the shortest path routing mechanism automatically provides functional backup paths for all origin-destination pairs that are still connected in the residual network and automatically restores the original routing paths as soon as the failing components become operational again, see Figure 3.5. As it takes some time to propagate the failure information through the network, recompute the shortest paths, and update forwarding tables accordingly, this restoration mechanism has a relatively high latency compared to methods based on pre-computed backup paths. It nevertheless is very popular in practice, because it requires no additional configuration overhead.

The most important drawback of shortest path routing is that it is usually less bandwidth efficient than other routing schemes. Because all traffic streams are sent along shortest paths with respect to the same global routing metric, there are strong interdependencies among the routing paths for

<sup>5)</sup>In fact, only the shortest possible address prefixes necessary to determine the outgoing link by a longest matching prefix search are stored in the forwarding tables. In order to further improve the forwarding efficiency, both protocols OSPF and IS-IS also have (slightly different) concepts of areas and hierarchies within an AS.



**Figure 3.5** Automatic restoration with shortest-path routing. The failing link is removed from the routers' network databases (or its length is implicitly set to infinity) and the shortest paths are recomputed in the residual network.

different traffic streams. Once two streams merge on their way to a common destination, they cannot be separated from each other anymore, for example.

Several questions that arise in the planning and operation of IP networks are concerned with the installation or the efficient use of network resources and lead to complex optimization problems. In the following, we describe the planning problems addressed in this thesis.

### 3.4 Optimization Problems in IP networks

One of the most basic basic planning task is the construction of the network itself. This **network design** task involves two types of decisions, those concerning the dimensioning of the network and those concerning the routing.

The **dimensioning** specifies what equipment is installed where in the network. Thereby, it also determines the network topology and the capacities available on the links. The planning of the dimensioning involves all decisions related to the installation of hardware, including which type of router and which interface cards to deploy at each node and which links to establish between the nodes, for example.

The **routing** specifies via which paths the data packets are sent through the network. For the routing within the network, this means to choose an appropriate metric, which then is supplied to the routing protocol for the shortest path computations. In principle, also the gateway routers' policies for the handling of external traffic must be determined. Due to the flexibility of the BGP protocol, however, this is more an issue of finding appropriate agreements with the peers than an issue of engineering and optimization.

The design of a network is a long-term strategic planning tasks. There-

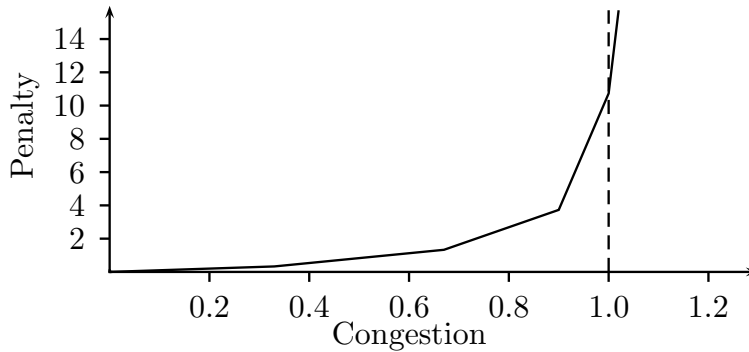
fore, the network should be constructed such that it meets with future traffic demands. Since these demands are usually unknown, the network's dimensioning and routing is typically planned on basis of a traffic forecast, which gives an estimate of the expected end-to-end traffic. Because of the TCP flow control mechanisms, individual IP traffic streams are extremely bursty on short time scale. Therefore, only average flows or demands within 1 to 5 minute intervals are considered in traffic accounting and forecasting in practice. Typically, forecasts based on the peak-traffic 5-minute interval or on the 95-percentile of all 5-minute intervals are used for network planning purposes.

Given a traffic forecast and the set of possible topology and hardware configurations, the task in the network design problem is to find a dimensioning and a corresponding routing metric, such that the resulting traffic flows do not exceed the provided capacities. The goal is to minimize the total cost of the dimensioning.

Depending on the actual routing protocol and the desired routing variant, several additional restrictions must be taken into account when choosing the routing metric. If each traffic stream shall be sent unsplit via single path, then the link metric must be chosen such that all shortest paths are uniquely determined. Otherwise it is unclear whether the traffic flows in the real network will exceed the link capacities or not. If ECMP is used, the metrics may be chosen such that there are multiple shortest paths between any pair of nodes, but the traffic splitting among these paths must be properly taken into account. Similarly, the network operator may allow an asymmetric metric or insist on a symmetric one for operational reasons. It is also very common to restrict the hop length or the geographical length of the routing paths in order to guarantee low transmission delays.

Often, operators wish to design their network such that it remains operational also in case of link or router failures. Even though the reconfiguration mechanisms of the dynamic shortest path routing protocols automatically establish backup routes in the remaining network, the impact of a failure may be severe. A routing metric that is optimal for the normal state, where all network elements are fully functional, may be inappropriate in a failure situation. In order to protect the network at least against those failures that statistically occur frequently, the dimensioning and the metric must be chosen in such a way, that the capacities suffice for the traffic flows not only in the fully operational network, but also in these failure scenarios.

If the dimensioning and the routing have been planned based on good traffic estimates, the users will typically observe good service quality in the newly deployed network. Over time, however, new applications and services will emerge and old ones will disappear. This may lead to rapid and unforeseen traffic changes, and the routing that would have been optimal for the predicted traffic demands then might provide very poor service quality for the actually observed traffic streams. The task of re-establishing good



**Figure 3.6** Link penalty function used by Fortz and Thorup [92, 93] and Ericsson et al. [81]. This function is a piecewise-linear approximation of the average packet delay on a link.

service in response to such traffic changes is called **traffic engineering**. Its goal is to find a new routing metric for the new (observed) traffic demands, such that the resulting flows fit into the capacities again or the service quality improves. The dimensioning of the network thereby remains unchanged. Traffic engineering is one of the most important short-term planning tasks in IP networks. By reconfiguring only the routing in the network, operators can react on changes in the traffic demands quickly and without costly changes to the hardware configuration.

Besides the packet loss-rate, the packet delay and the jitter (i.e., the variation of the packet delays) are the most important measures for the service quality in IP networks. Modern real-time streaming applications, such as audio or video broadcasts or interactive network games, require low and nearly constant packet delays. Objectives like minimizing the average packet delay or minimizing the average packet loss-rate are commonly used in traditional traffic engineering. Using standard results from queuing theory, these objective functions can be (approximately) expressed as the sum over all links of a penalty function that depends on the link congestion. The penalty (i.e., the packet loss-rate or delay) on a single link is small as long as the congestion is low, but it rapidly increases if the flow gets close to the capacity, see Figure 3.6. Fortz and Thorup [92, 93], Ericsson et al. [81], and others use such an objective function for traffic engineering in the IP network of AT&T, for example.

Another commonly used objective in traffic engineering is to minimize the maximum congestion over all links in the network. This yields a so-called max-min fair routing with respect to all three quality measures loss-rate, delay, and jitter. The factor by which all demands can be increased simultaneously without exceeding the given capacities is maximal for such a routing. Deploying a max-min fair routing thus not only improves the service quality in the network, it also makes the network more robust against unforeseen future traffic changes.

When the traffic demands change too much and the existing network does



not provide adequate capacities for any routing anymore, a **redimensioning** of the network becomes necessary. The network operator must decide how to modify the current dimensioning in order to satisfy the new traffic demands. Both the installation of new node devices or links as well as the removal of no longer needed ones are possible options.

Traditionally, the objective in network redimensioning is to minimize the expenses that are necessary to provide sufficient capacities again. Given the current network dimensioning and a traffic forecast, the goal is to find both a new dimensioning and a new routing, such that the traffic flows do not exceed the new capacities and the cost of the expansion is minimized.

Alternatively, network redimensioning can be considered as a generalization of traffic engineering, where a given budget can be spent to change the dimensioning of the network. In this case, the goal is to find a new dimensioning and a new routing, such that the given reconfiguration budget is not exceeded and the traffic engineering objective is optimized.

In practice, it is impossible to change the dimensioning of the entire network at once. Typically, operational needs lead to restrictions concerning the number of new installations, removals, or reconfigurations of hardware elements, which must be considered in the network redimensioning planning. Also, the costs of the transition to the new network design involve not only the cost for new devices or links but also the expenses of carrying out the reconfiguration. In particular, there may be costs for the removal or reconfiguration of devices or links.

### 3.5 Mathematical Model

As we have seen in the previous section, building and operating an IP network involves various complex planning tasks. In this section, we introduce the notation that is necessary to describe these tasks mathematically as optimization problems. We restrict our attention to those technical variants that arise in the planning of backbone networks and confine our definitions to the traditional single shortest path routing version without traffic splitting, which is used in the German national research and education network DFN. In order to keep the mathematical notation as neat as possible, we will also simplify the real-world hardware installation possibilities and ignore several operational routing constraints, such as path length restrictions through most parts of this thesis. Mathematical models including all these real-world technical and operational details are presented in Chapter 9 at the end of this thesis.

As the traffic demands and flows in IP networks typically are highly asymmetric and, furthermore, most routing protocols admit the configuration of asymmetric end-to-end routings, the network can be modeled adequately

only in a directed context.

**Notation 3.1** *The (potential) topology of the communication network is modeled as a directed **supply graph**  $D = (V, A)$ .*

The nodes of the supply graph correspond to the locations where a router is or may be installed, its arcs represent all existing or potentially installable directed links between these routers. At most one router may be installed at each node. Multiple (independently configurable) routers at one location need to be modeled by multiple nodes. Broadcast link technologies connecting more than two routers via a single shared capacity link are very unusual in high speed backbone networks. Therefore, they are not considered in this thesis. Without loss of generality we may assume that  $D$  is connected and contains no loops.

Although the use of multicast applications such as video and audio broadcast is steadily increasing, the fraction of real multicast traffic in nowadays IP networks is still negligible. For the sake of simplicity, we therefore also ignore multicast traffic in this thesis and assume that every communication demand has a single source and a single destination node.

**Notation 3.2** *The traffic demands in the network are given as a set of directed **commodities**  $K \subseteq V \times V$ . Each commodity  $(s, t) \in K$  has an associated **demand value**  $d_{s,t} \in \mathbb{Z}_+$ . The nodes  $s$  and  $t$  are called the **source** and the **target** (or **destination**) of commodity  $(s, t)$ , respectively. Both  $s$  and  $t$  together are called the **terminals** of  $(s, t)$ .*

Note that we implicitly assume in this definition that there are no parallel commodities, i.e.,  $K \subseteq V \times V$ . As we are interested in unsplittable shortest path routings only, this assumption can be made without loss of generality. If there were two parallel commodities from  $s$  to  $t$  with demand values  $d_1$  and  $d_2$ , then both commodities would be routed along the same unique shortest  $(s, t)$ -path in any unsplittable shortest path routing. So, they can be aggregated into one commodity with demand value  $d_1 + d_2$ . Furthermore, we may assume without loss of generality that  $K$  contains no loop commodities  $(v, v)$ , that  $d_{s,t} > 0$  for all  $(s, t) \in K$ , and that there exists at least one  $(s, t)$ -path in  $D$  for each  $(s, t) \in K$ .

Let  $\mathcal{P}$  denote the set of all simple directed paths in  $D$ . The set of all simple directed  $(s, t)$ -paths between two nodes  $s, t \in V$  is denoted by  $\mathcal{P}(s, t)$ . For the set of all paths among the node pairs of a commodity set  $K \subseteq V \times V$ , we write  $\mathcal{P}(K) := \bigcup_{(s,t) \in K} \mathcal{P}(s, t)$ .

In general, a (multi-path) routing is an assignment of the demand values of the commodities to end-to-end transportation paths. For each commodity, it determines which fraction of the demand is sent via which path from the source to the destination. Within an IP routing domain, these paths are computed by some shortest path routing protocol according to the routing

lengths assigned to the link interfaces. In the digraph  $D = (V, A)$ , these routing weights correspond to arc lengths.

**Notation 3.3** A *(routing) metric* <sup>6)</sup> is a vector  $\lambda = (\lambda_a)_{a \in A} \in \mathbb{R}_+^A$ . The individual arc lengths  $\lambda_a$  are also called **routing lengths**.

In practice, the routing lengths must be small integers that fit into the data format of the routing protocol. Depending on the specific protocol, the range of admissible values for individual link lengths or for the total length of the resulting end-to-end paths is bounded. The OSPF protocol, for example, admits only routing lengths up to  $2^{16} - 1 = 65535$ . In principle, it thus might happen that a shortest path routing that is given by some unbounded real-valued routing metric  $\lambda \in \mathbb{R}_+^A$  cannot be realized by the real-world routing protocol in practice. In modern routing protocols, however, the ranges of admissible routing lengths are sufficiently large. All possible shortest path routings can be realized in a network of realistic size and the range restriction thus can be safely ignored. We address this issue in detail in Chapter 4.

Once the routing metric is set, the routers autonomously compute the shortest paths. If there were multiple equally-long shortest paths between some routers, then it depends on the version and the actual implementation of the routing protocol whether traffic is split or not and how ambiguities are resolved. To obtain a well-defined unsplittable shortest path routing, the metric must be chosen such that the shortest path is unique for each commodity. If a metric  $\lambda \in \mathbb{R}_+^A$  defines a *unique* shortest path from  $s$  to  $t$  in the digraph  $D$ , then (and only then) we denote this path by  $P_{(s,t)}^*(\lambda)$ . For brevity, we also write  $P_{(s,t)}^*$  if it is clear which metric  $\lambda$  we refer to.

### Definition 3.4

(i) A routing metric  $\lambda \in \mathbb{R}_+^A$  is said to be **valid** for the commodity set  $K \subseteq V \times V$  if, for each commodity  $(s, t) \in K$ , the shortest  $(s, t)$ -path  $P_{(s,t)}^*(\lambda)$  is uniquely determined.

(ii) Given a valid metric  $\lambda \in \mathbb{R}_+^A$  for the commodity set  $K$ , the path set

$$S(\lambda) := \{ P_{(s,t)}^*(\lambda) : (s, t) \in K \}$$

is the **unsplittable shortest path routing (USPR)** induced by  $\lambda$  for  $K$ .

Note that a valid metric needs to induce unique shortest  $(s, t)$ -paths only for those node pairs  $(s, t)$  that comprise commodities. For all other node pairs, ambiguous shortest paths are allowed. Nevertheless, each valid metric

---

<sup>6)</sup>Throughout this thesis, we will use the term *metric* for any vector of routing lengths, independent of whether or not the triangle inequality holds. Although mathematically imprecise, this notion is the de-facto standard in the IP routing literature.

can be easily turned into a metric that induces unique shortest paths not only for the given commodities but for all node pairs.

Given a valid metric or its induced USPR, the total traffic flows across the arcs can be easily computed.

**Notation 3.5** Let  $\lambda \in \mathbb{R}_+^A$  be a valid metric for the commodity set  $K$ . The **induced flow** (or **total flow**) on arc  $a$  then is

$$f_a(\lambda) := \sum_{(s,t) \in K: a \in P_{(s,t)}^*(\lambda)} d_{s,t} \quad \text{for each } a \in A. \quad (3.1)$$

For a given a path set  $S \subseteq \mathcal{P}(K)$  containing exactly one  $(s, t)$ -path for each commodity  $(s, t) \in K$ , we define the induced arcs flows equivalently as

$$f_a(S) := \sum_{P \in S: a \in P} d_{s_P, t_P} \quad \text{for each } a \in A. \quad (3.2)$$

If  $\lambda \in \mathbb{R}_+^A$  is a valid metric for the commodity set  $K$  and  $S$  is its induced USPR, then we clearly have  $f_a(\lambda) = f_a(S)$  for all  $a \in A$ .

With this notation, we now can formalize the three network design and routing planning problems that we consider throughout this thesis.

One of the most common short-term planning tasks for IP networks is traffic engineering. Its goal is to improve the service quality of an existing network by (re-)optimizing the routing of the traffic demands, while leaving the network's topology and hardware configuration unchanged. An important measure for the service quality is the link congestion. The loss-rate, the average delay, and the jitter (the variation of delays) highly depend on the congestion of the links the packets traverse. A typical objective in traffic engineering therefore is to minimize the maximum link congestion in the network.

Mathematically, this task can be formulated as the minimum congestion unsplittable shortest path routing problem, denoted briefly as MIN-CON-USPR. Given a digraph  $D = (V, A)$  with fixed arc capacities  $u_a \in \mathbb{Z}_+$ ,  $a \in A$ , and a set of commodities  $K \subseteq V \times V$  with demand values  $d_{s,t} \in \mathbb{Z}_+$ ,  $(s, t) \in K$ , the task is to find a metric  $\lambda \in \mathbb{Z}_+^A$  that defines an USPR for the given commodity set  $K$  and minimizes the maximum congestion  $f_a(\lambda)/u_a$  over all arcs.

<i>Problem:</i>	MIN-CON-USPR
<i>Instance:</i>	A digraph $D = (V, A)$ with arc capacities $u_a \in \mathbb{Z}_+$ , $a \in A$ , and a commodity set $K \subseteq V \times V$ with demands $d_{s,t} \in \mathbb{Z}_+$ , $(s, t) \in K$ .
<i>Solution:</i>	A valid metric $\lambda \in \mathbb{Z}_+^A$ , i.e., a metric $\lambda$ such that the shortest $(s, t)$ -path w.r.t. $\lambda$ is unique for each $(s, t) \in K$ .
<i>Objective:</i>	$\min\{L \in \mathbb{R}_+ : f_a(\lambda) \leq L u_a \text{ for all } a \in A\}$ .

An important strategic planning task is to design or to redimension a network such that it can accommodate some forecasted traffic demands. This involves decisions about the network topology, the dimensioning and configuration of the router and link hardware, and the routing of the traffic demands within the planned network. The objective typically is to minimize (an estimate of) the total network costs, which are assumed to be the sum of all installed hardware component costs plus maybe some fixed charge costs accounting for the set-up of the nodes and links. In Chapter 9 we present a mathematical model for the network design task that captures the possible hardware configurations in IP networks at a very detailed level. Until then, however, we restrict our attention to only two very basic cases — the capacitated network design problem CAP-USPR and the fixed charge network design problem FC-USPR.

In both problems, we are given a digraph  $D = (V, A)$  with arc capacities  $u_a \in \mathbb{Z}_+$  and arc costs  $w_a \in \mathbb{Z}_+$  for all  $a \in A$  and a set of commodities  $K \subseteq V \times V$  with demand values  $d_{s,t} \in \mathbb{Z}_+$  for all  $(s, t) \in K$ . In the capacitated network design problem with unsplittable shortest path routing, called CAP-USPR, the given capacities  $w_a$  are interpreted as basic capacity units that can be installed in integer multiplicities on the arcs. In this problem, we seek non-negative integer capacity multipliers  $z_a \in \mathbb{Z}_+$  for all arcs  $a \in A$  and a valid metric  $\lambda \in \mathbb{Z}_+^A$  for the commodities  $K$  (i.e., a metric inducing a unique shortest  $(s, t)$ -path in  $D$  for each  $(s, t) \in K$ ), such that the induced arc flows  $f_a(\lambda)$  do not exceed the installed capacities  $u_a z_a$ . The objective is to minimize the total capacity installation cost  $\sum_{a \in A} w_a z_a$ .

<i>Problem:</i>	CAP-USPR
<i>Instance:</i>	A digraph $D = (V, A)$ with arc capacities $u_a \in \mathbb{Z}_+$ , $a \in A$ , and arc costs $w_a \in \mathbb{Z}_+$ , $a \in A$ , and a commodity set $K \subseteq V \times V$ with demands $d_{s,t} \in \mathbb{Z}_+$ , $(s, t) \in K$ .
<i>Solution:</i>	Capacity multipliers $z \in \mathbb{Z}_+^A$ and a valid metric $\lambda \in \mathbb{Z}_+^A$ , such that $f_a(\lambda) \leq u_a z_a$ for all $a \in A$ .
<i>Objective:</i>	$\min \sum_{a \in A} z_a w_a$ .

CAP-USPR is one of the simplest non-trivial network design problems with unsplittable shortest path routing that allows the installation of arbitrarily large capacities.

In the fixed charge network design problem with unsplittable shortest path routing FC-USPR, the given capacities are regarded as fix. Here the task is to find a minimum cost arc set  $B \subseteq A$  and a valid metric  $\lambda \in \mathbb{Z}_+^B$ , such that  $\lambda$  defines an USPR for the commodities  $K$  within the subgraph  $(V, B)$  and the induced arc flows  $f_a(\lambda)$  do not exceed the capacities  $u_a$  (on the arcs  $a \in B$ ). This problem is the simplest version of a network design problem with unsplittable shortest path routing admitting only bounded capacities.

<i>Problem:</i>	FC-USPR
<i>Instance:</i>	A digraph $D = (V, A)$ with arc capacities $u_a \in \mathbb{Z}_+$ , $a \in A$ , and arc costs $w_a \in \mathbb{Z}_+$ , $a \in A$ , and a commodity set $K \subseteq V \times V$ with demands $d_{s,t} \in \mathbb{Z}_+$ , $(s, t) \in K$ .
<i>Solution:</i>	An arc set $B \subseteq A$ and a valid metric $\lambda \in \mathbb{Z}_+^B$ , such that <ul style="list-style-type: none"> <li>(i) for each <math>(s, t) \in K</math>, <math>\lambda</math> induces a unique shortest <math>(s, t)</math>-path <math>P_{(s,t)}^*(\lambda)</math> in the subgraph <math>(V, B)</math> and</li> <li>(ii) <math>f_a(\lambda) \leq u_a</math> for all <math>a \in B</math>.</li> </ul>
<i>Objective:</i>	$\min \sum_{a \in B} w_a$ .

In all three problems MIN-CON-USPR, CAP-USPR, and FC-USPR, we may assume without loss of generality that  $u_a > 0$  for all  $a \in A$  and  $d_{s,t} > 0$  for all  $(s, t) \in K$ .

In Part II of this thesis, we analyze the computational complexity of these three problems and present polynomial time approximation algorithms for several special cases. An exact (but non-polynomial) integer programming solution approach for these problems is developed in Part III. In this approach, we decompose the problem of finding an unsplittable shortest path routing for the given commodity set into the two subproblems of finding the end-to-end routing paths for the commodities and, afterwards, finding a valid metric that induces exactly these paths. This approach requires a good (algorithmic) characterization of those end-to-end routings that indeed correspond to unsplittable shortest paths routings and (practically) efficient methods for finding a valid metric afterwards. These two aspects are discussed in the following.

## Part I

# Metrics and Routing Paths





## Chapter 4

# The Inverse Unique Shortest Paths Problem

In this chapter, we study INVERSE UNIQUE SHORTEST PATHS problems that arise in the context of shortest path routing planning. Given a collection of paths in a directed graph  $D = (V, A)$ , the general task is to find a compatible metric for the paths, i.e., a metric of non-negative arc lengths such that the given paths are uniquely determined shortest paths between their respective terminals. In the basic problem version IUSP we admit arbitrary fractional arc lengths, which is equivalent to allowing arbitrarily large integer arc lengths. This problem corresponds to the task of deciding whether the given routing paths can be realized at all with the abstract shortest path routing paradigm, ignoring all possible obstacles of a routing protocol implementation. In the second problem version MIN-PATH-IUSP we seek for an integer-valued compatible metric that minimizes the length of the longest of the prescribed paths. Similarly, the third problem version MIN-ARC-IUSP seeks for an integer-valued compatible metric where the length of the longest arc is minimal. The later two problems MIN-PATH-IUSP and MIN-ARC-IUSP arise in practice, if the prescribed routing paths shall be realized with some real, implemented shortest path routing protocol, such as OSPF, IS-IS, PNNI, or RIP. Each of these protocols admits only a bounded range of arc and/or path lengths.

In Section 4.2 we formally introduce the three IUSP problems. After a review of the related literature in Section 4.1, we show in Section 4.3 that the problem of finding a fractional compatible metric (or proving its non-existence) is solvable in polynomial time by linear programming techniques.

In Section 4.4, the two problem variants MIN-PATH-IUSP and MIN-ARC-IUSP seeking for small integer lengths are proven to be  $\mathcal{APX}$ -hard. More precisely, we show that the minimal longest path length cannot be approximated within a factor less than  $8/7$  and the minimal longest arc length not within a factor less than  $9/8$ , unless  $\mathcal{P} = \mathcal{NP}$ . This answers the (previously) open question whether these problems are  $\mathcal{NP}$ -hard or not. Furthermore,

these inapproximability results imply that it is  $\mathcal{NP}$ -hard to decide whether a prescribed set of paths can be realized with any real shortest path routing protocol such as OSPF, IS-IS, PNNI, or RIP.

A simple algorithm that achieves an  $\mathcal{O}(|V|)$ -approximation guarantee for both integer-valued IUSP variants by rounding the optimal solution of the corresponding linear programming relaxations is presented in Section 4.5.

In Section 4.6 we finally discuss a variant of the INVERSE UNIQUE SHORTEST PATHS problem where the potential routing is not given in terms of its end-to-end paths but in terms of the arcs that lead from each node towards the different destinations – similar to the forwarding tables used by the network routers in practice. This alternative representation is very useful if the potential routing is given in terms of the forwarding tables anyway, or if not the complete end-to-end paths but only some of the flow-arcs shall be prescribed, as in intermediate steps of a routing optimization process, for example.

## 4.1 Related Work

Computing shortest paths in a weighted digraph is one of the classical combinatorial optimization problems. Various algorithms are known to solve this problem efficiently. The inverse problem of finding arc lengths that induce a prescribed set of paths or node-to-node distances has received only little attention in the mathematical literature.

Several groups studied the inverse shortest paths problem in the context of data engineering where the task is to reconstruct data from inaccurate measurements or observations. Typical applications are the estimation of the average travel times on road segments from total end-to-end travel times, the recovery of the densities of earth crust layers from observed seismic waves, or the reconstruction of relationship degrees in genetic sequencing [49, 142]. In these settings, the goal is to find edge or arc lengths that match as closely as possible the observed distances and shortest paths. Neither the uniqueness of the observed shortest paths nor the integrality of the lengths are required in these problem variants. Burton and Toint [51, 50] apply a quadratic programming approach to estimate the arc lengths when the perceived shortest paths are known. For the same problem, Tong and Lam [185] propose a conjugate gradient method. The case with additional upper bounds on the shortest paths' lengths is discussed by Burton et al. [52]. Fekete et al. [89] study the complexity of a problem variant where the lengths are to be reconstructed from observed distances in the network, but the shortest paths are not known. Cai and Li [53] show that the more general INVERSE MATROID INTERSECTION problem can be transformed into a minimum cost flow problem and can be solved in strongly polynomial time.

In a similar application context, Ahuja and Orlin [2, 3] study the INVERSE

LINEAR PROGRAMMING problem, which subsumes the INVERSE NETWORK FLOW and the INVERSE SHORTEST PATH problem; see also Zhang et al. [195, 198]. Given a linear programming problem  $\min\{c^T x : Ax \leq b\}$ , the task is to find a perturbed cost vector  $d$  such that a prescribed solution  $x^*$  is optimal with respect to  $d$  and the  $L_1$  or the  $L_\infty$  norm of  $c - d$  is minimized. Ahuja and Orlin show that both inverse problem variants can be formulated as linear programming problems, too, and can be solved efficiently with linear programming or combinatorial techniques. However, neither the uniqueness of the prescribed optimal solution nor the integrality of the perturbed cost vector are addressed.

Inverse shortest path problems with integer lengths and unique shortest paths has been rarely addressed. Farago et al. [85, 86] study a special case where the given paths are known to be shortest paths with respect to the number of edges and where the task is to find lengths such that all these paths are unique shortest paths. Ben-Ameur and Gourdin [27] discuss structural properties of (undirected) path sets where all paths are uniquely determined shortest paths for some common edge lengths. Ben-Ameur and Gourdin present several linear programming formulations for finding fractional edge lengths that are compatible with the given paths. Based on these formulations, they also propose heuristics for finding small integer edge lengths that are compatible with the given paths, but they leave open whether the problem of finding the smallest possible such lengths is  $\mathcal{NP}$ -hard or not. The approximation algorithm for MIN-PATH-IUSP and MIN-ARC-IUSP we present in Section 4.5 and the linear programming formulation for IUSP we present in the following are (variants of) those proposed by Ben-Ameur and Gourdin [27].

## 4.2 Problem Definition

A major difficulty in planning of shortest path networks is to find a link metric that induces a set of globally efficient end-to-end routing paths. In this chapter, we are concerned with the problem of finding a metric that induces a prescribed set of paths. In addition, we typically want the metric to satisfy the technical restrictions of the used routing protocol, mainly limiting the range of the admissible routing lengths. Given a collection of paths in a directed graph, we wish to find small positive integer arc lengths for which the prescribed paths are uniquely determined shortest paths between their respective terminals. This problem naturally arises when the end-to-end routing paths are provided by a prior dimensioning or routing optimization process, or if the paths are predetermined by some technical or operational reasons.

The uniqueness of the prescribed shortest paths as well as the range of admissible length values are very important concerns for shortest path rout-

ing in telecommunications. The routers in a network calculate the shortest paths autonomously. Thus, the metric must be chosen such that for any pair of nodes *only* the prescribed paths are shortest paths. Otherwise, the traffic flows in the real network might differ from those computed during network planning. For unsplittable shortest path routing, this means that each prescribed path must be the uniquely determined shortest path between its terminals.

In practice, the arc lengths furthermore must be small integers that fit into the data format of the routing protocol. For link-state protocols like OSPF or IS-IS, this means that each single link length is bounded by some protocol-dependent constant. With a distance-vector protocol such as RIP, the total length of each routing path is bounded. Real network routing domains typically range from 10 to 1000 nodes. Thus, the limited size of the lengths typically is not relevant for OSPF, which admits link lengths between 1 and  $2^{16}-1$ . However, it becomes an important issue for (the initial version of) IS-IS, which admits values up to 63 only. For RIP (Release 1), where the link lengths must be chosen such that the total length of the longest routing path does not exceed 15, the restricted range of admissible lengths is one of the major difficulties in practice. Therefore, it is natural to seek integer lengths that either minimize the largest link length or the length of the longest path.

Let  $D = (V, A)$  be a directed graph. We denote the set of all simple paths in  $D$  by  $\mathcal{P}$  and the set of all simple  $(s, t)$ -paths between two nodes  $s$  and  $t$  by  $\mathcal{P}(s, t)$ . Given a directed path  $P$ , we refer to its source node by  $s_P$  and to its target node by  $t_P$ . By  $P[u, v]$  we denote the  $(u, v)$ -subpath of a path  $P$ . For notational simplicity, we also write  $\lambda(P) := \sum_{a \in P} \lambda_a$  for any metric  $\lambda = (\lambda_a)_{a \in A} \in \mathbb{R}_+^A$  and any path  $P \in \mathcal{P}$ .

Throughout this section, we are not concerned about some specific commodity set or its associated demand values. In a very general setting, we simply say that a path set  $S \subseteq \mathcal{P}$  comprises a unique shortest path system if and only if there exists a metric such that each path in  $S$  is the unique shortest path between its terminals.

#### Definition 4.1

- (i) A metric  $\lambda = (\lambda_a)_{a \in A} \in \mathbb{R}_+^A$  is said to be **compatible** with a given path set  $S \subseteq \mathcal{P}$ , if each path  $P \in S$  is the unique shortest  $(s_P, t_P)$ -path with respect to  $\lambda$ .
- (ii) A path set  $S \subseteq \mathcal{P}$  is a **unique shortest path system (USPS)** if there exists a compatible metric  $\lambda \in \mathbb{R}_+^A$  for  $S$ . Otherwise  $S$  is called a **non-USPS**.

Clearly, each unsplittable shortest path routing  $S$  for a specific commodity set  $K \subseteq V \times V$  is a USPS. Vice versa, each USPS  $S$  defines an unsplittable

shortest path routing for the commodity set  $K := \{(s_P, t_P) : P \in S\}$  and each metric that is compatible with  $S$  is valid for the commodity set  $K$ .

The most basic problem related to unique shortest path systems is to decide whether some path set  $S \subseteq \mathcal{P}$  is a USPS and, if so, to find *some* compatible metric. Formally, this INVERSE UNIQUE SHORTEST PATHS problem is given as follows:

<i>Problem:</i>	IUSP
<i>Instance:</i>	A digraph $D = (V, A)$ and a path set $S \subseteq \mathcal{P}$ .
<i>Task:</i>	Find a metric $\lambda \in \mathbb{R}_+^A$ that is compatible with $S$ or prove that none exists.

The problem of finding an integer-valued metric that uniquely induces the prescribed shortest paths and minimizes the length of the longest of these paths can be formally defined as follows:

<i>Problem:</i>	MIN-PATH-IUSP
<i>Instance:</i>	A digraph $D = (V, A)$ and a path set $S \subseteq \mathcal{P}$ .
<i>Solution:</i>	Integer metric $\lambda \in \mathbb{Z}_+^A$ with $\lambda_a \geq 1$ for all $a \in A$ that is compatible with $S$ .
<i>Objective:</i>	$\min \max\{\lambda(P) : P \in S\}$ .

The problem variant, where we wish to minimize the maximum arc length, is given as:

<i>Problem:</i>	MIN-ARC-IUSP
<i>Instance:</i>	A digraph $D = (V, A)$ and a path set $S \subseteq \mathcal{P}$ .
<i>Solution:</i>	Integer metric $\lambda \in \mathbb{Z}_+^A$ with $\lambda_a \geq 1$ for all $a \in A$ that is compatible with $S$ .
<i>Objective:</i>	$\min \max\{\lambda_a : a \in A\}$ .

The IUSP problem corresponds to task of finding some maybe fractional (or, equivalently, arbitrarily large integer) arc lengths that realize the prescribed routing paths with an abstract unsplittable shortest path routing protocol. The MIN-PATH-IUSP problem corresponds to the task of finding admissible arc lengths for a real distance-vector protocol that induce the given paths in practice, while MIN-ARC-IUSP corresponds to the same task for a link-state protocol.

Without loss of generality, we can assume in both problems MIN-PATH-IUSP and MIN-ARC-IUSP that the given path set  $S$  is a USPS, i.e., there exists at least some metric that is compatible with  $S$ . With the linear programming approach presented in Section 4.3, it can be decided in polynomial time whether this is the case or not.

All results presented in the following also hold for the undirected problem versions, where we seek for edge lengths that uniquely induce a given set of

prescribed shortest paths or destination-arc pairs in an undirected graph.

### 4.3 Finding Real-Valued Lengths

Suppose we are given a path set  $S \subseteq \mathcal{P}$  and we wish to find some maybe fractional metric  $\lambda \in \mathbb{R}_+^A$  such that each path  $P \in S$  is the unique shortest  $(s_P, t_P)$ -path with respect to  $\lambda$ , or prove that no such metric exists. As a path  $P$  is the unique shortest  $(s_P, t_P)$ -path if and only if  $\lambda(P) < \lambda(P')$  for all  $P' \in \mathcal{P}(s_P, t_P) \setminus \{P\}$ , this problem is equivalent to the problem of finding a solution for the following linear inequality system:

$$\sum_{a \in P'} \lambda_a - \sum_{a \in P} \lambda_a > 0 \quad \forall P \in S, P' \in \mathcal{P}(s_P, t_P) \setminus \{P\} \quad (4.1)$$

$$\lambda_a \geq 0 \quad \forall a \in A \quad (4.2)$$

**Observation 4.2** *A path set  $S \subseteq \mathcal{P}$  is a USPS if and only if the linear inequality system (4.1)–(4.2) has a solution.*

For each path set  $S \subseteq \mathcal{P}$ , the set of metrics that are compatible with  $S$  form an open polyhedral cone: A metric  $\lambda \in \mathbb{R}_+^A$  is compatible with  $S$  if and only if, for any  $c > 0$ , the metric  $c\lambda = (c\lambda_a)_{a \in A}$  is compatible with  $S$  and, furthermore, any convex combination of two metrics that both are compatible with  $S$  is compatible with  $S$ , too. The faces of this cone are defined by the inequalities (4.1) and (4.2). As the faces corresponding to the strict inequalities (4.1) are not fully contained within this cone, one cannot (naively) apply a standard linear programming algorithm for finding a compatible metric for  $S$  or proving its non-existence.

Yet, as (for now) we are only interested in finding *some* metric  $\lambda$  that is compatible with  $S$ , we may tighten the strict inequalities (4.1) to some non-strict inequalities, provided that the entire linear system remains feasible after the tightening if it was before. In fact, one easily observes that any metric that is compatible with a given path set  $S$  can be transformed into a metric with strictly positive and integer arc lengths that is compatible with  $S$ , too, by scaling and rounding.

**Proposition 4.3** *A path set  $S \subseteq \mathcal{P}$  is a USPS if and only if there exists a compatible metric  $\lambda \in \mathbb{Z}_+^A$  with  $\lambda_a \geq 1$  for all  $a \in A$ .*

**Proof.** If there there exists a metric with strictly positive integer arc lengths that is compatible with  $S$ , then  $S$  is a USPS by definition.

To prove the other implication, let  $S \subseteq \mathcal{P}$  be a USPS and let  $\lambda \in \mathbb{R}_+^A$  be an arbitrary metric that is compatible with  $S$ . We define

$$\alpha(\lambda) := \min\{\lambda(P') - \lambda(P) : P \in S, P' \in \mathcal{P}(s_P, t_P) \setminus \{P\}\} .$$

Since each path  $P \in S$  is the unique shortest  $(s_P, t_P)$ -path with respect to  $\lambda$ , we have  $\alpha(\lambda) > 0$ . Multiplying all arc lengths by the factor of  $|V|/\alpha(\lambda)$ , we obtain the metric  $\lambda' \in \mathbb{R}_+^A$  with  $\lambda'_a := |V|/\alpha(\lambda) \cdot \lambda_a$  for all  $a \in A$ , which satisfies

$$\alpha(\lambda') = \min\{\lambda'(P') - \lambda'(P) : P \in S, P' \in \mathcal{P}(s_P, t_P) \setminus \{P\}\} \geq |V|.$$

This means that, for each path  $P \in S$ , the length of the prescribed shortest  $(s_P, t_P)$ -path  $P$  and of the second shortest  $(s_P, t_P)$ -path with respect to  $\lambda'$  differ by at least  $|V|$ . As no simple path contains more than  $|V| - 1$  arcs, each path  $P \in S$  will remain the shortest  $(s_P, t_P)$ -path even if we round up all arc length of this metric to an integer and set all zero lengths to one. More formally, the metric  $\lambda'' \in \mathbb{Z}_+^A$  given by

$$\lambda''_a := \begin{cases} 1, & \text{if } \lambda'_a = 0, \text{ and} \\ \lceil \lambda'_a \rceil, & \text{otherwise} \end{cases}$$

is compatible with  $S$ . □

Proposition 4.3 implies that unique shortest path systems could have been defined equivalently by requiring the existence of a compatible metric with only strictly positive and integer arc lengths instead of non-negative fractional ones. Furthermore, it immediately leads to a linear programming formulation of the IUSP problem with only non-strict linear inequalities.

**Theorem 4.4 (Ben-Ameur and Gourdin [27])** *A path set  $S \subseteq \mathcal{P}$  is a USPS if and only if the linear inequality system*

$$\sum_{a \in P'} \lambda_a - \sum_{a \in P} \lambda_a \geq 1 \quad \forall P \in S, P' \in \mathcal{P}(s_P, t_P) \setminus \{P\}, \quad (4.3)$$

$$\lambda_a \geq 1 \quad \forall a \in A \quad (4.4)$$

*has a solution. Furthermore, any solution  $\lambda \in \mathbb{R}_+^A$  of (4.3)–(4.4) is a compatible metric for  $S$ .*

**Proof.** Follows directly from Proposition 4.3. □

Since (4.3)–(4.4) contains only non-strict inequalities, standard linear programming techniques can be applied to find a solution or prove that none exists. In general, however, this linear system may contain exponentially many (non-redundant) inequalities of type (4.3). In order to solve it efficiently, we must solve the following separation problem:

<i>Problem:</i>	SEP-IUSP
<i>Instance:</i>	A digraph $D = (V, A)$ , a path set $S \subseteq \mathcal{P}$ , and a metric $\lambda \in \mathbb{R}_+^A$ .
<i>Task:</i>	Find two paths $P \in S$ and $P' \in \mathcal{P}(s_P, t_P) \setminus \{P\}$ , such that (4.3) is violated for $P$ and $P'$ , or to prove that no such paths exist.

The problem SEP-IUSP can be solved in polynomial time by algorithm `CheckTwoShortestPaths` illustrated at the bottom of this page. For each  $P \in S$ , algorithm `CheckTwoShortestPaths` computes the two shortest  $(s_P, t_P)$ -paths  $P^1$  and  $P^2$  with respect to  $\lambda$  and compares their respective lengths with the length of  $P$ .

**Lemma 4.5 (Ben-Ameur and Gourdin [27])** *Algorithm `CheckTwoShortestPaths` solves the separation problem SEP-IUSP in polynomial time.*

**Proof.** Let  $P \in S$  and consider the shortest and the second shortest  $(s_P, t_P)$ -path  $P^1$  and  $P^2$ , respectively.

If  $P$  is the shortest path and the length of the second shortest path  $P^2$  is greater or equal to  $\lambda(P) + 1$ , then inequality (4.3) clearly holds for  $P$  and all other  $(s_P, t_P)$ -paths  $P' \neq P$  as well.

If this is not the case, then we have  $P \neq P^1$  or  $\lambda(P^2) < \lambda(P) + 1$ . If  $P \neq P^1$ , i.e., the actual shortest  $(s_P, t_P)$ -path w.r.t.  $\lambda$  is not the prescribed shortest  $(s_P, t_P)$ -path  $P$ , then the inequality  $\lambda(P^1) - \lambda(P) \geq 1$  is obviously violated. Otherwise, if  $P = P^1$  but  $\lambda(P^2) < \lambda(P) + 1$ , then the inequality  $\lambda(P^2) - \lambda(P) \geq 1$  is violated, which concludes the proof of the correctness of algorithm `CheckTwoShortestPaths`.

It is also very easy to see that algorithm `CheckTwoShortestPaths` runs polynomial in the size of the given problem instance. It performs at most  $|S|$

---

#### Algorithm 4.1 `CheckTwoShortestPaths`

---

**Input:** Digraph  $D = (V, A)$ , path set  $S \subseteq \mathcal{P}$ , and metric  $\lambda \in \mathbb{R}_+^A$

**Output:** A pair  $(P, P')$  for which (4.3) is violated or assertion that no such pair exists.

1. For each  $P \in S$  do
    2. Compute the shortest and the second shortest  $(s_P, t_P)$ -paths  $P^1$  and  $P^2$  with respect to  $\lambda$  (with Yen's algorithm [196] f.e.)
      - 2.1 If  $P^1 \neq P$  then
 

Return  $(P, P' := P^1)$ .
      - 2.2 If  $P^1 = P$  and  $\lambda(P^2) < \lambda(P^1) + 1$  then
 

Return  $(P, P' := P^2)$ .
  - 3 Return 'All inequalities (4.3) are satisfied.'
-



many 2-shortest path computations, and each of these computations can be performed in polynomial time using any of the  $k$ -shortest paths algorithms of Yen [196], Katoh et al. [122], Azevedo et al. [12], Eppstein [78], Martins and Santos [139], or Skiscim and Golden [178], for example.  $\square$

It follows from the polynomial time equivalence of separation and optimization that one can find a solution or prove the infeasibility of (4.3)–(4.4) in polynomial time using a standard cutting plane algorithm, see Grötschel et al. [103], Grötschel and Lovász [101], or Wolsey [194]. With Theorem 4.4 we immediately obtain the following theorem.

**Theorem 4.6 (Ben-Ameur and Gourdin [27])** *The IUSP problem is solvable in polynomial time.*

Another linear programming formulation for the IUSP problem with only polynomially many variables and inequalities is discussed in Section 5.7.

If the given path set  $S$  is a USPS, then any solution of (4.3)–(4.4) is a compatible metric for  $S$ . Using Cramer’s rule (or some variant of the scaling and rounding technique used in the proof of Proposition 4.3), any such metric can be easily transformed into a compatible metric with integer values. However, the arc lengths produced by these approaches may be extremely large. In the following section we show that the problems MIN-ARC-IUSP and MIN-PATH-IUSP of finding a compatible metric with integer arc lengths as small as possible are computationally hard.

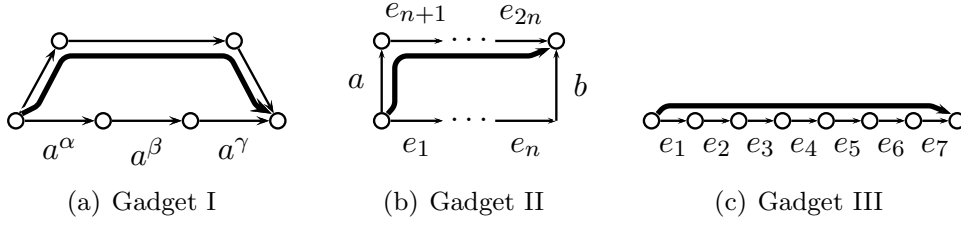
## 4.4 Inapproximability Results for Integer Lengths

In the first part of this section, we show that the problem MIN-PATH-IUSP is  $\mathcal{APX}$ -hard.

**Theorem 4.7** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate MIN-PATH-IUSP within a factor of  $8/7 - \epsilon$ .*

**Proof.** We construct a polynomial reduction from the  $\mathcal{NP}$ -complete decision problem SET PARTITION(3) to the problem of computing a solution of value strictly less than 8 for MIN-PATH-IUSP. SET PARTITION(3) is a restricted variant of the SET PARTITION problem introduced by Karp [121] (also called EXACT SET COVER by Garey and Johnson [96]). Given a finite set  $I$  and a collection  $\mathcal{C} \subset 2^I$  with  $|\{C \in \mathcal{C} : i \in C\}| = 3$  for all  $i \in I$ , the task is to decide whether there exists a subcollection  $\mathcal{C}' \subseteq \mathcal{C}$  which forms a partition of  $I$ . In SET PARTITION(3), each element occurs in exactly three sets of the collection. It was shown by Papadimitriou and Yannakakis [156] that this restricted variant of SET PARTITION remains  $\mathcal{NP}$ -complete.

Suppose we are given an instance of SET PARTITION(3) that consists of the set  $I$  and the collection  $\mathcal{C} = \{C_j \subset I : j \in J\}$ . For each element  $i \in I$ , we



**Figure 4.1** Gadgets used in the proof.

let  $J(i) := \{\alpha(i), \beta(i), \gamma(i)\} = \{j \in J : i \in C_j\}$  be the indices of the three sets in  $\mathcal{C}$  that contain  $i$ . We construct a MIN-PATH-IUSP instance consisting of a digraph  $D = (V, A)$  and a shortest path system  $S$  that consists of the three basic gadgets illustrated in Figure 4.1. Each gadget corresponds to a small digraph containing a single prescribed shortest path. Gadget I is used to force the length of at least one of the arcs  $a^\alpha$ ,  $a^\beta$ , and  $a^\gamma$  to two (or more) by prescribing a unique shortest path of only three arcs that is parallel to the path  $(a^\alpha, a^\beta, a^\gamma)$ . Gadget II is used to ‘propagate’ the length of some arc  $a$  to some other arc  $b$ . If  $\lambda_{e_i} = 1$  for all  $i$  in the graph of Gadget II, then  $\lambda_b \geq \lambda_a$ . Gadget III is used to ensure  $\lambda_{e_i} = 1$  for all  $i$  in any solution where the longest prescribed path length is 7.

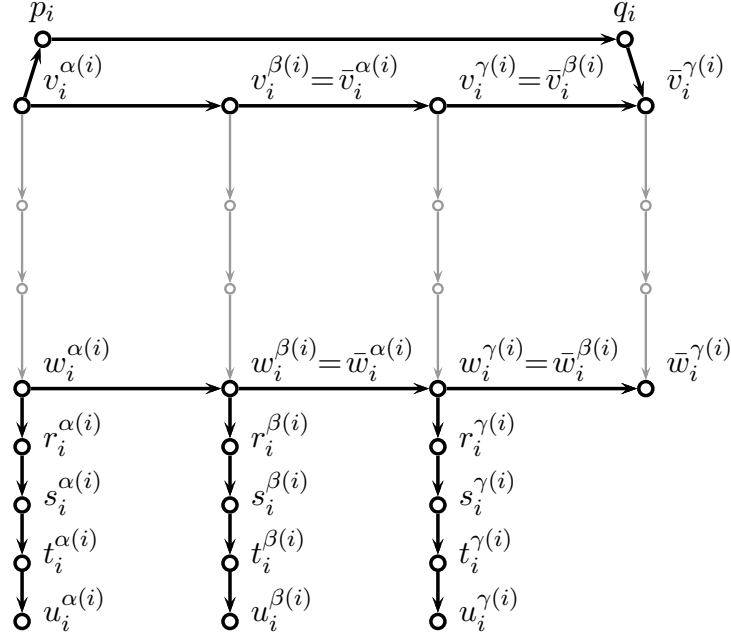
The idea of our construction is the following. For each element  $i \in I$ , we prescribe two shortest paths. The first path is parallel to the arcs  $a_i^{\alpha(i)}$ ,  $a_i^{\beta(i)}$ , and  $a_i^{\gamma(i)}$ , as in Gadget I. These three arcs correspond to the three sets  $C_{\alpha(i)}$ ,  $C_{\beta(i)}$ , and  $C_{\gamma(i)}$  that contain  $i$ . The unique shortest path requirement forces the length of at least one of these arcs to two (or more), which corresponds to the condition that  $i$  is contained in at least one set of a feasible partition. The second prescribed shortest path for  $i$  consists of three arcs  $b_i^{\alpha(i)}$ ,  $b_i^{\beta(i)}$ , and  $b_i^{\gamma(i)}$ . Also these arcs correspond to the three sets containing  $i$ . Several gadgets of types II and III are used to ensure that the lengths of  $b_i^j$  and  $a_i^j$  are chosen coherently for all elements  $i$  and sets  $j$  such that  $\lambda_{b_i^j} \geq \lambda_{a_i^j} + 1$ . By requiring that the length of the second prescribed shortest path  $(b_{\alpha(i)}, b_{\beta(i)}, b_{\gamma(i)})$  does not exceed 7, we then can enforce that element  $i$  is contained in at most one set of a feasible collection.

We construct the digraph  $D = (V, A)$  as follows. For each element  $i \in I$ , we introduce 22 nodes

$$V_i := \{r_i^j, s_i^j, t_i^j, u_i^j, v_i^j, w_i^j : j \in J(i)\} \cup \{p_i, q_i, \bar{v}_i^{\gamma(i)}, \bar{w}_i^{\gamma(i)}\}.$$

Some of these nodes receive multiple names. For every  $i \in I$ , we also denote  $\bar{v}_i^{\alpha(i)} := v_i^{\beta(i)}$ ,  $\bar{v}_i^{\beta(i)} := v_i^{\gamma(i)}$ ,  $\bar{w}_i^{\alpha(i)} := w_i^{\beta(i)}$ , and  $\bar{w}_i^{\beta(i)} := w_i^{\gamma(i)}$ . Among these nodes, we add the 21 arcs

$$\begin{aligned} A_i := & \{(v_i^j, \bar{v}_i^j), (w_i^j, \bar{w}_i^j), (w_i^j, r_i^j), (r_i^j, s_i^j), (s_i^j, t_i^j), (t_i^j, u_i^j) : j \in J(i)\} \\ & \cup \{(v_i^{\alpha(i)}, p_i), (p_i, q_i), (q_i, \bar{v}_i^{\gamma(i)})\}. \end{aligned}$$



**Figure 4.2** Nodes  $V_i$  and arcs  $A_i$  introduced for any  $i \in I$  in bold.

These nodes and arcs are illustrated in Figure 4.2.

We introduce further nodes and arcs for pairs of elements that occur together in some set of the collection. For every set  $C_j \in \mathcal{C}$  and every ordered pair  $(i, k) \in C_j^2$ , we introduce the nodes and the arcs

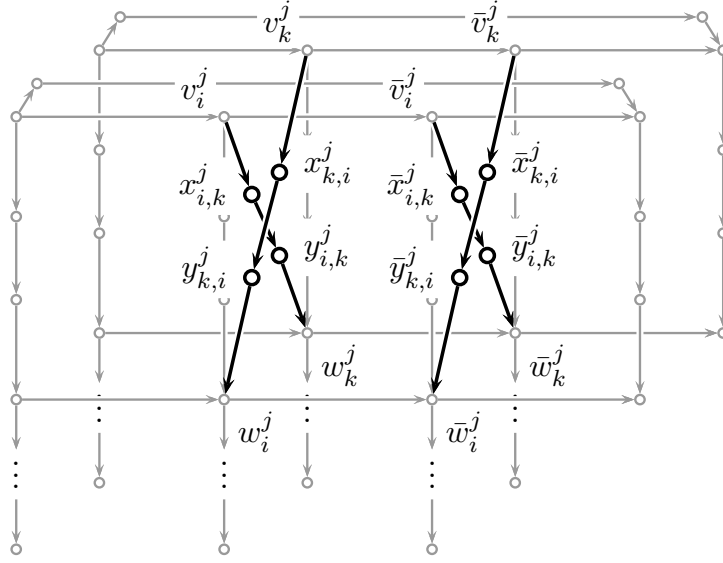
$$V_{i,k}^j := \{x_{i,k}^j, \bar{x}_{i,k}^j, y_{i,k}^j, \bar{y}_{i,k}^j\} \quad \text{and} \\ A_{i,k}^j := \{(v_i^j, x_{i,k}^j), (x_{i,k}^j, y_{i,k}^j), (y_{i,k}^j, w_k^j), (\bar{v}_i^j, \bar{x}_{i,k}^j), (\bar{x}_{i,k}^j, \bar{y}_{i,k}^j), (\bar{y}_{i,k}^j, \bar{w}_k^j)\}$$

as illustrated in Figure 4.3. For pairs  $(i, k)$  with  $i \neq k$ , we may assume without loss of generality that  $i$  and  $k$  occur together in only one set  $C_j \in \mathcal{C}$ . For the pairs  $(i, i)$  of identical elements, which do occur ‘together’ in the three sets  $C_{\alpha(i)}$ ,  $C_{\beta(i)}$ , and  $C_{\gamma(i)}$ , we identify the vertices  $\bar{x}_{i,i}^\alpha(i) = x_{i,i}^{\beta(i)}$ ,  $\bar{x}_{i,i}^\beta(i) = x_{i,i}^{\gamma(i)}$ ,  $\bar{y}_{i,i}^\alpha(i) = y_{i,i}^{\beta(i)}$ , and  $\bar{y}_{i,i}^\beta(i) = y_{i,i}^{\gamma(i)}$ . This ensures that there are a unique  $(v_i^j, w_k^j)$ -path and a unique  $(\bar{v}_i^j, \bar{w}_k^j)$ -path in the constructed digraph for all  $C_j \in \mathcal{C}$  and  $i, k \in C_j$ ; see Figure 4.5.

Together, these sets form the simple digraph  $D = (V, A)$  with

$$V := \bigcup_{i \in I} V_i \cup \bigcup_{j \in J, (i,k) \in C_j^2} V_{i,k}^j \quad \text{and} \quad A := \bigcup_{i \in I} A_i \cup \bigcup_{j \in J, (i,k) \in C_j^2} A_{i,k}^j.$$

Figure 4.2 illustrates the subgraph of all nodes and arcs that are introduced for a single element  $i \in I$ . Figures 4.3 and 4.4 illustrate nodes and arcs that are introduced for the pairs  $(i, k) \in C_j^2$  with  $i \neq k$ , and Figure 4.5 illustrates the nodes and arcs that are introduced for pairs  $(i, i) \in C_j^2$ .



**Figure 4.3** Nodes  $V_{i,k}^j \cup V_{k,i}^j$  and arcs  $A_{i,k}^j \cup A_{k,i}^j$  introduced for a pair  $(i, k) \in \mathcal{C}_j^2$  with  $i \neq k$  and  $j = \beta(i) = \beta(k)$  in bold.

In this digraph  $D$ , we prescribe various shortest paths. For each single element  $i \in I$ , we prescribe the two paths

$$S_i := \{ (v_i^{\alpha(i)}, p_i, q_i, \bar{v}_i^{\gamma(i)}), (w_i^{\alpha(i)}, w_i^{\beta(i)}, w_i^{\gamma(i)}, \bar{w}_i^{\gamma(i)}) \}.$$

For each ordered pair  $(i, k) \in I^2$ , with  $i, k \in C_j$  for some  $C_j \in \mathcal{C}$  (including the pairs with  $i = k$ ), we prescribe the two paths

$$S_{i,k}^j := \{ (v_i^j, \bar{v}_i^j, \bar{x}_{i,k}^j, \bar{y}_{i,k}^j, \bar{w}_k^j), (v_i^j, x_{i,k}^j, y_{i,k}^j, w_k^j, r_k^j, s_k^j, t_k^j, u_k^j) \}.$$

Figures 4.6 and 4.7 illustrate these paths. The set of all prescribed paths is

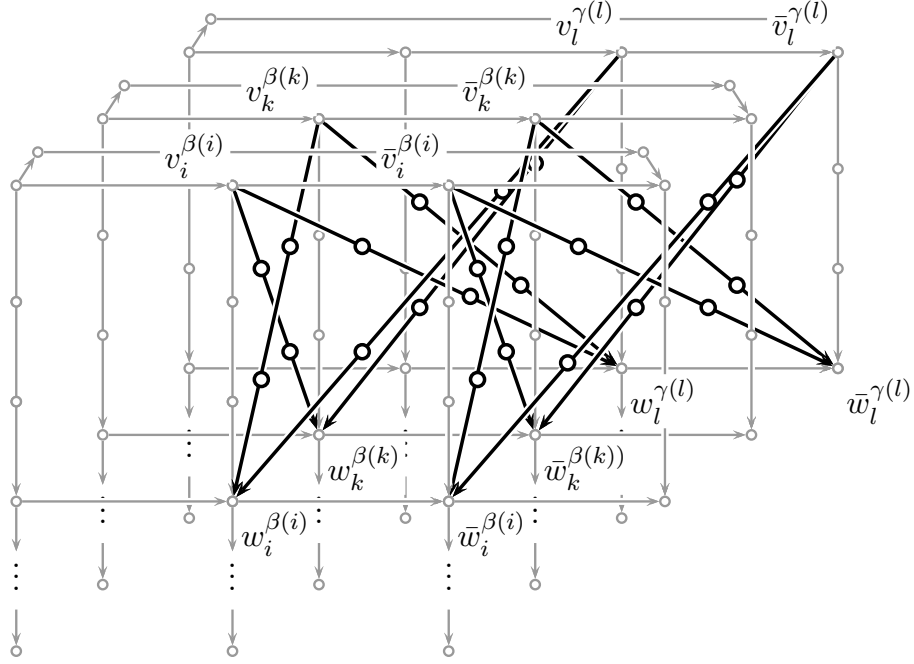
$$S := \bigcup_{i \in I} S_i \cup \bigcup_{j \in J, (i,k) \in \mathcal{C}_j^2} S_{i,k}^j.$$

It is obvious that this construction is polynomial.

In the first part of this proof, we show that the constructed MIN-PATH-IUSP instance is solvable for any given SET PARTITION(3) instance, i.e.,  $S$  is a USPS. Furthermore, we show that the value of any solution of the MIN-PATH-IUSP instance is at least 7. Consider the arc lengths  $\lambda \in \mathbb{Z}_+^A$  defined as

$$\lambda_a := \begin{cases} 3, & \text{for all } a = (w_i^j, \bar{w}_i^j) \text{ with } j \in J \text{ and } i \in C_j, \\ 2, & \text{for all } a = (v_i^j, \bar{v}_i^j) \text{ with } j \in J \text{ and } i \in C_j, \\ 2, & \text{for all } a = (p_i, q_i) \text{ with } i \in I, \text{ and} \\ 1, & \text{otherwise.} \end{cases}$$

It is straightforward to verify that each path in  $S$  is the uniquely determined shortest path between its terminals for these lengths:



**Figure 4.4** All nodes and arcs introduced for pairs of different elements in a set  $C_j = \{i, k, l\} \in \mathcal{C}$  with  $j = \beta(i) = \beta(k) = \gamma(l)$  in bold, i.e., the nodes  $V_{i,i}^j \cup V_{k,k}^j \cup V_{l,l}^j \cup V_{i,k}^j \cup V_{k,i}^j \cup V_{i,l}^j \cup V_{l,i}^j \cup V_{k,l}^j \cup V_{l,k}^j$  and the arcs  $A_{i,i}^j \cup A_{k,k}^j \cup A_{l,l}^j \cup A_{i,k}^j \cup A_{k,i}^j \cup A_{i,l}^j \cup A_{l,i}^j \cup A_{k,l}^j \cup A_{l,k}^j$ .

Let  $P = (v_i^{\alpha(i)}, p_i, q_i, \bar{v}_i^{\gamma(i)})$  for some  $i \in I$ . The only other  $(v_i^{\alpha(i)}, \bar{v}_i^{\gamma(i)})$ -path in  $D$  is  $P' = (v_i^{\alpha(i)}, v_i^{\beta(i)}, v_i^{\gamma(i)}, \bar{v}_i^{\gamma(i)})$ . Since  $\lambda(P) = 4 < 6 = \lambda(P')$ , the path  $P$  is the unique shortest  $(v_i^{\alpha(i)}, \bar{v}_i^{\gamma(i)})$ -path w.r.t.  $\lambda$ .

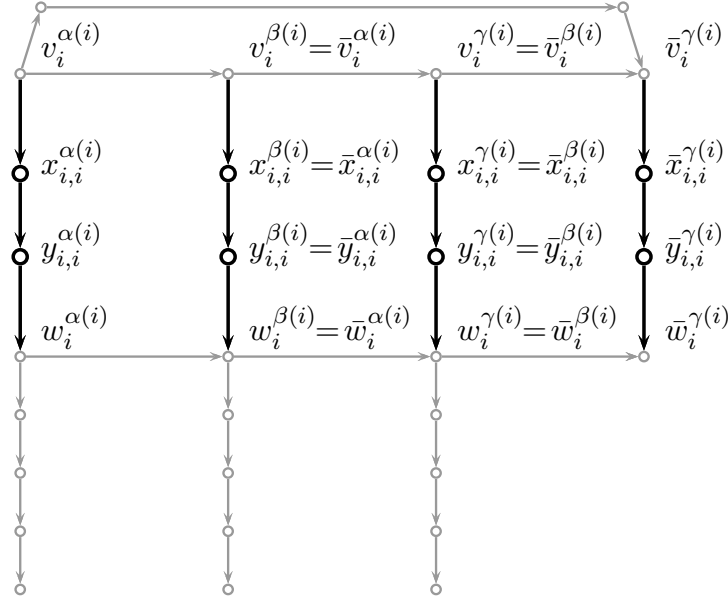
Let  $P = (w_i^{\alpha(i)}, w_i^{\beta(i)}, w_i^{\gamma(i)}, \bar{w}_i^{\gamma(i)})$  for  $i \in I$ . Since  $P$  is the only  $(w_i^{\alpha(i)}, \bar{w}_i^{\gamma(i)})$ -path in  $D$ , is also is the uniquely determined shortest one.

Let  $P = (v_i^j, \bar{v}_i^j, \bar{x}_{i,k}^j, \bar{y}_{i,k}^j, \bar{w}_k^j)$  for some  $j \in J$  and some pair  $(i, k) \in C_j^2$  (including the pairs with  $i = k$ ). The only other  $(v_i^j, \bar{w}_k^j)$ -path in  $D$  is  $P' = (v_i^j, x_{i,k}^j, y_{i,k}^j, w_k^j, \bar{w}_k^j)$ . Because  $\lambda(P) = 5 < 6 = \lambda(P')$ , the path  $P$  is the unique shortest  $(v_i^j, \bar{w}_k^j)$ -path w.r.t.  $\lambda$ .

Finally, let  $P = (v_i^j, x_{i,k}^j, y_{i,k}^j, w_k^j, r_k^j, s_k^j, t_k^j, u_k^j)$  for some  $j \in J$  and  $(i, k) \in C_j^2$ . As  $P$  is the only path from  $v_i^j$  to  $u_k^j$ , it also is the uniquely determined shortest one.

Since each path in  $S$  is the unique shortest path between its terminals, the lengths  $\lambda$  are compatible with  $S$ .

Let  $P = (v_i^j, x_{i,k}^j, y_{i,k}^j, w_k^j, r_k^j, s_k^j, t_k^j, u_k^j)$  for some  $j \in J$  and  $(i, k) \in C_j^2$ . As  $P \in S$  and  $|P| = 7$ , we have  $\max\{\lambda(P) : P \in S\} \geq 7$  for all arc lengths that are compatible with  $S$  and satisfy  $\lambda_a \geq 1$  for all  $a \in A$ .



**Figure 4.5** All nodes  $V_{i,i}^{\alpha(i)} \cup V_{i,i}^{\beta(i)} \cup V_{i,i}^{\gamma(i)}$  and arcs  $A_{i,i}^{\alpha(i)} \cup A_{i,i}^{\beta(i)} \cup A_{i,i}^{\gamma(i)}$  introduced for any  $i \in I$  in bold.

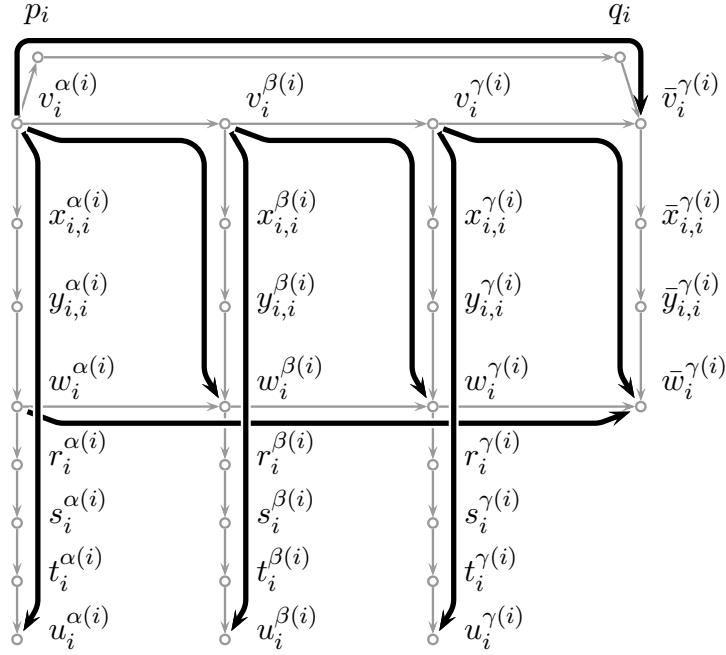
In the second part of the proof, we show that the constructed MIN-PATH-IUSP instance has a solution of value 7 if the given SET PARTITION(3) instance is solvable. Assume there is a subcollection  $\mathcal{C}' \subseteq \mathcal{C}$  that forms a partition of  $I$ . We define the arc lengths  $\lambda = \lambda(\mathcal{C}') \in \mathbb{Z}_+^A$  as

$$\lambda_a := \begin{cases} 3, & \text{for all } a = (w_i^j, \bar{w}_i^j) \text{ with } C_j \in \mathcal{C}' \text{ and } i \in C_j, \\ 2, & \text{for all } a = (w_i^j, \bar{w}_i^j) \text{ with } C_j \notin \mathcal{C}' \text{ and } i \in C_j, \\ 2, & \text{for all } a = (v_i^j, \bar{v}_i^j) \text{ with } C_j \in \mathcal{C}' \text{ and } i \in C_j, \text{ and} \\ 1, & \text{otherwise.} \end{cases} \quad (4.5)$$

Analogous to the first part, we now verify that each path in  $S$  is a unique shortest path between its terminals and its length is at most 7. For this, we again consider each of the four path types in  $S$  individually.

Let  $P = (v_i^{\alpha(i)}, p_i, q_i, \bar{v}_i^{\gamma(i)})$  for some  $i \in I$ . By (4.5), we have  $\lambda_a = 1$  for all  $a \in P$ , and thus  $\lambda(P) = 3$ . According to our construction, the only other  $(v_i^{\alpha(i)}, \bar{v}_i^{\gamma(i)})$ -path in  $D$  is  $P' = (v_i^{\alpha(i)}, v_i^{\beta(i)}, v_i^{\gamma(i)}, \bar{v}_i^{\gamma(i)})$ . Since  $\mathcal{C}'$  defines a partition, the element  $i$  is contained in some set  $C_j \in \mathcal{C}'$ . By (4.5), this implies  $\lambda_{(v_i^j, \bar{v}_i^j)} = 2$  and, therefore,  $\lambda(P') \geq 4$ .

Let  $P = (w_i^{\alpha(i)}, w_i^{\beta(i)}, w_i^{\gamma(i)}, \bar{w}_i^{\gamma(i)})$  for  $i \in I$ . As  $\mathcal{C}'$  defines a partition of  $I$ , the element  $i$  is contained in exactly one set  $C_j \in \mathcal{C}'$ . W.l.o.g., suppose this set is  $C_{\alpha(i)}$ . According to (4.5), we then have  $\lambda_{(w_i^{\alpha(i)}, \bar{w}_i^{\alpha(i)})} = 3$  and  $\lambda_{(w_i^{\beta(i)}, \bar{w}_i^{\beta(i)})} = \lambda_{(w_i^{\gamma(i)}, \bar{w}_i^{\gamma(i)})} = 2$ . Thus,  $\lambda(P) = 7$ . Since  $P$  is the unique  $(w_i^{\alpha(i)}, \bar{w}_i^{\gamma(i)})$ -path in  $D$ , it also is the unique shortest such path.



**Figure 4.6** Prescribed shortest paths  $S_i \cup S_{i,i}^{\alpha(i)} \cup S_{i,i}^{\beta(i)} \cup S_{i,i}^{\gamma(i)}$  for any  $i \in I$ .

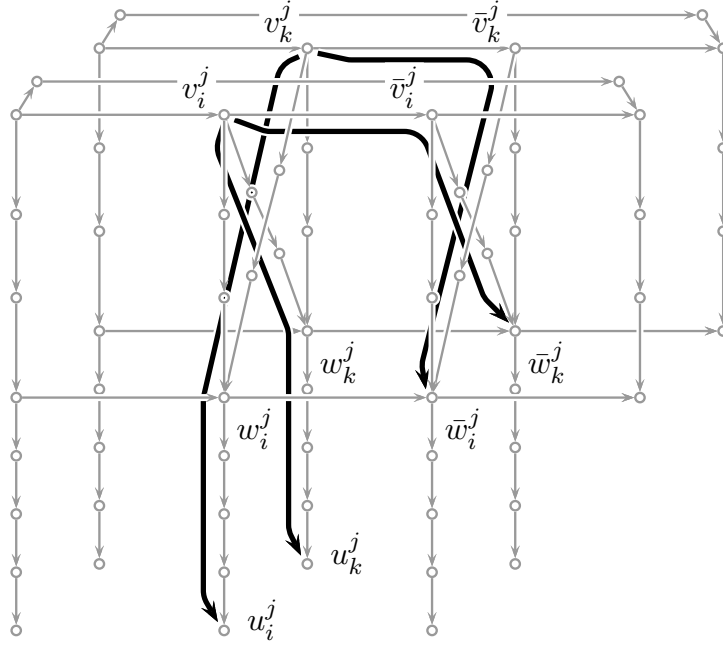
Let  $P = (v_i^j, \bar{v}_i^j, \bar{x}_{i,k}^j, \bar{y}_{i,k}^j, \bar{w}_k^j)$  for  $j \in J$  and  $(i, k) \in C_j^2$  (including pairs with  $i = k$ ). If  $C_j \in \mathcal{C}'$ , then (4.5) implies  $\lambda_{(v_i^j, \bar{v}_i^j)} = 2$ , otherwise  $\lambda_{(\bar{v}_i^j, \bar{v}_i^j)} = 1$ . All other arcs in  $P$  have a length of one. Consequently,  $\lambda(P) \leq 5$ . The only other  $(v_i^j, \bar{w}_k^j)$ -path in  $D$  is  $P' = (v_i^j, x_{i,k}^j, y_{i,k}^j, w_k^j, \bar{w}_k^j)$ . If  $C_j \in \mathcal{C}'$ , then  $\lambda_{(v_i^j, \bar{v}_i^j)} = 2$  and  $\lambda_{(w_k^j, \bar{w}_k^j)} = 3$ . Otherwise,  $\lambda_{(v_i^j, \bar{v}_i^j)} = 1$  and  $\lambda_{(w_k^j, \bar{w}_k^j)} = 2$ . Independent of whether  $C_j \in \mathcal{C}'$  or not, it follows that  $\lambda(P) < \lambda(P')$ .

Finally, let  $P = (v_i^j, x_{i,k}^j, y_{i,k}^j, w_k^j, r_k^j, s_k^j, t_k^j, u_k^j)$  for some  $j \in J$  and  $(i, k) \in C_j^2$ . By (4.5), all arcs in  $P$  have length one. Hence,  $\lambda(P) = 7$ . Furthermore,  $P$  is the solitary  $(v_i^j, k_k^j)$ -path.

Summarizing these arguments, it follows that the arc lengths  $\lambda = \lambda(\mathcal{C}')$  are compatible with  $S$  and that  $\max\{\lambda(P) : P \in S\} = 7$ . This concludes the second part of the proof.

In the following third part, it remains to show that the constructed MIN-PATH-IUSP instance has a solution of value 7 *only* if the given SET PARTITION(3) instance is solvable. For this, we prove that any arc length vector  $\lambda \in \mathbb{Z}_+^a$  with  $\lambda_a \geq 1$  for all  $a \in A$  that is compatible with  $S$  and satisfies  $\max\{\lambda(P) : P \in S\} = 7$  can be transformed into a subcollection  $\mathcal{C}' \subseteq \mathcal{C}$  that defines a partition of  $I$ . So, suppose we are given such arc lengths and define the subcollection  $\mathcal{C}' = \mathcal{C}'(\lambda)$  as

$$\mathcal{C}' := \{C_j \in \mathcal{C} : \lambda_{(v_i^j, \bar{v}_i^j)} \geq 2 \text{ for some } i \in C_j\}. \quad (4.6)$$



**Figure 4.7** Prescribed shortest paths  $S_{i,k}^j \cup S_{k,i}^j$  for some elements  $i, k \in I$  with  $i \neq k$  and  $j = \beta(i) = \beta(k)$ .

Note that  $P = (v_i^j, x_{i,k}^j, y_{i,k}^j, w_k^j, r_k^j, s_k^j, t_k^j, u_k^j) \in S$  for all  $j \in J$  and  $(i, k) \in C_j^2$ . Since  $|P| = 7$ , all arcs in  $P$  must have length 1. In particular,

$$\lambda_{(v_i^j, x_{i,k}^j)} = \lambda_{(x_{i,k}^j, y_{i,k}^j)} = \lambda_{(y_{i,k}^j, w_k^j)} = 1 \quad \text{for all } j \in J \text{ and } (i, k) \in C_j^2 \quad (4.7)$$

First, we show that  $\mathcal{C}'$  covers all elements in  $I$ . Let  $i \in I$  and consider the two  $(v_i^{\alpha(i)}, \bar{v}_i^{\gamma(i)})$ -paths  $P = (v_i^{\alpha(i)}, p_i, q_i, \bar{v}_i^{\gamma(i)})$  and  $P' = (v_i^{\alpha(i)}, v_i^{\beta(i)}, v_i^{\gamma(i)}, \bar{v}_i^{\gamma(i)})$ . Since  $P \in S$  and the lengths  $\lambda$  are compatible with  $S$ , the inequality  $\lambda(P') \geq \lambda(P) + 1$  must hold. With  $\lambda_a \geq 1$  for all  $a \in A$ , it follows that  $\lambda(P') \geq 4$ . This, in turn, implies  $\lambda_{(v_i^j, \bar{v}_i^j)} \geq 2$  for some  $j \in J(i)$ . According to (4.6),  $C_j$  then belongs to  $\mathcal{C}'$  and the element  $i$  is covered by  $\mathcal{C}'$ .

Secondly, we show that the sets in  $\mathcal{C}'$  are pairwise disjoint. Suppose  $\mathcal{C}'$  contains two non-disjoint sets  $C_j$  and  $C_k$ ,  $j \neq k$ . Then there exists an element  $i \in C_j \cap C_k$ . Since  $C_j$  belongs to  $\mathcal{C}'$ , there must be some  $h \in C_j$  with  $\lambda_{(v_h^j, \bar{v}_h^j)} \geq 2$ . For this  $h$ , consider the two  $(v_h^j, \bar{w}_i^j)$ -paths  $P_1 = (v_h^j, \bar{v}_h^j, \bar{x}_{h,i}^j, \bar{y}_{h,i}^j, \bar{w}_i^j)$  and  $P_2 = (v_h^j, x_{h,i}^j, y_{h,i}^j, w_i^j, \bar{w}_i^j)$ . As  $\lambda$  is compatible with  $S$  and  $P_1 \in S$ , the lengths  $\lambda$  must satisfy  $\lambda(P_2) \geq \lambda(P_1) + 1$ . With (4.7) and  $\lambda_{(v_h^j, \bar{v}_h^j)} \geq 2$ , this implies  $\lambda_{(w_i^j, \bar{w}_i^j)} \geq 3$ . Analogously,  $C_k \in \mathcal{C}'$  implies  $\lambda_{(w_i^k, \bar{w}_i^k)} \geq 3$ . For the third index  $l$  with  $\{j, k, l\} = J(i)$ , a similar argument yields  $\lambda_{(w_i^l, \bar{w}_i^l)} \geq 2$ .

Now consider the path  $P = (w_i^{\alpha(i)}, w_i^{\beta(i)}, w_i^{\gamma(i)}, \bar{w}_i^{\gamma(i)}) \in S$ . Since all arcs  $(w_i^j, \bar{w}_i^j)$ ,  $(w_i^k, \bar{w}_i^k)$ , and  $(w_i^l, \bar{w}_i^l)$  are contained in  $P$ , we have  $\lambda(P) \geq 8$ .



This, however, conflicts with our assumption  $\max\{\lambda(P) : P \in S\} = 7$ . Consequently, all sets in  $\mathcal{C}'$  are pairwise disjoint and form a partition of  $I$ .

Together, the three parts of the proof imply that our construction maps any given SET PARTITION(3) instance to a solvable MIN-PATH-IUSP instance. Furthermore, the optimum solution value of the constructed MIN-PATH-IUSP instance is 7 if and only if the given SET PARTITION(3) instance is solvable, and is at least 8 otherwise. As SET PARTITION(3) is  $\mathcal{NP}$ -complete and the above construction is polynomial, it is  $\mathcal{NP}$ -hard to approximate MIN-PATH-IUSP within a factor strictly less than  $8/7$ .  $\square$

The same inapproximability bound of  $8/7$  can be shown for the corresponding undirected problem variant of MIN-PATH-IUSP: Let  $G = (V, E)$  be the undirected graph obtained by replacing each arc  $(v, w)$  of  $D = (V, A)$  with an undirected edge  $vw$  and consider  $S$  as a set of undirected paths in  $G$  in the above construction. For any arc length vector  $\lambda \in \mathbb{Z}_+^A$  examined for the directed graph  $D$ , now consider the corresponding undirected edge length vector  $\lambda^U \in \mathbb{Z}_+^E$  with  $\lambda_{uv}^U := \lambda_{(u,v)}$  for all  $(u, v) \in A$ . One easily verifies that the undirected paths in  $S$  are unique shortest paths between their respective terminals also in the undirected graph  $G$  with respect to the corresponding undirected lengths constructed in the first and in the second part of the proof. The third part of the proof carries over literally to the undirected case.

In the remainder of this section, we show that also the second problem MIN-ARC-IUSP is  $\mathcal{APX}$ -hard. This follows in a straightforward manner from the constant inapproximability threshold for MIN-PATH-IUSP.

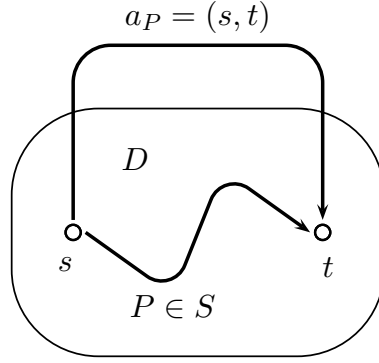
**Theorem 4.8** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate MIN-ARC-IUSP within a factor of  $9/8 - \epsilon$ .*

**Proof.** Suppose we are given a MIN-PATH-IUSP instance consisting of a digraph  $D = (V, A)$  and a shortest path system  $S$ . Consider the MIN-ARC-IUSP instance with the same path set  $S$ , but in an extended digraph  $D^* := (V, A \cup A^*)$  with  $A^* := \{a_P = (s, t) : P \in S, P \text{ is an } (s, t)\text{-path}\}$ , as illustrated in Figure 4.8: In  $D^*$ , we introduce a new arc  $a_P = (s, t)$  for each  $(s, t)$ -path in  $S$ , even if there already exists a parallel arc  $(s, t)$  in  $A$ .

Clearly, the constructed MIN-ARC-IUSP instance is solvable. Furthermore, for any arc length vector  $\lambda \in \mathbb{Z}_+^A$  that is compatible with  $S$  in  $D$ , the arc lengths  $\lambda^* = (\lambda_a^*)_{a \in A \cup A^*} \in \mathbb{Z}_+^{A \cup A^*}$  defined as

$$\lambda_a^* := \begin{cases} \lambda_a & , \text{ for all } a \in A, \text{ and} \\ \lambda(P) + 1 & , \text{ for all } a_P \in A^*, \end{cases}$$

are compatible with  $S$  in  $D^*$ . Conversely, all arc lengths  $\lambda^* \in \mathbb{Z}_+^{A \cup A^*}$  in  $D^*$  that are compatible with  $S$  define arc lengths  $\lambda = (\lambda_a^*)_{a \in A}$  in  $D$  that are compatible with  $S$  and satisfy  $\lambda(P) \leq \lambda_{a_P}^* - 1$  for all  $P \in S$ .



**Figure 4.8** Extension of digraph  $D$  to digraph  $D^*$ .

As it is  $\mathcal{NP}$ -hard to find lengths  $\lambda$  in  $D$  such that the longest path in  $S$  is strictly less than 8, it is also  $\mathcal{NP}$ -hard to find lengths  $\lambda^*$  in  $D^*$  such that all arc lengths (in particular those of the arcs in  $A^*$ ) are strictly less than 9. Therefore, it is  $\mathcal{NP}$ -hard to approximate MIN-ARC-IUSP within a factor less than  $9/8$ .  $\square$

Analogously, one can show the same inapproximability threshold of  $9/8$  for the corresponding undirected problem.

## 4.5 An LP-Rounding Algorithm

A path  $P$  is the unique shortest  $(s, t)$ -path with respect to some integer arc lengths  $\lambda \in \mathbb{R}_+^A$  with  $\lambda_a \geq 1$  for all  $a \in A$  if and only if  $\lambda(P') - \lambda(P) \geq 1$  for all  $P' \in \mathcal{P}(s, t) \setminus \{P\}$ . This leads to the following integer linear programming formulation for MIN-ARC-IUSP:

$$\begin{aligned}
 & \min \lambda_{\max} && \text{(ARC-IUSP)} \\
 & \sum_{a \in P'} \lambda_a - \sum_{a \in P} \lambda_a \geq 1 && \forall P \in S, P' \in \mathcal{P}(s_P, t_P) \setminus \{P\} \quad (4.8) \\
 & 1 \leq \lambda_a \leq \lambda_{\max} && \forall a \in A \\
 & \lambda_a \in \mathbb{Z}_+ && \forall a \in A
 \end{aligned}$$

The correctness of this formulation follows immediately from Theorem 4.4.

Formulation (ARC-IUSP) contains exponentially many (non-redundant) inequalities of type (4.8), but these inequalities can be separated in polynomial time using algorithm `CheckTwoShortestPaths` presented in Section 4.3, for example. Hence, the linear programming relaxation of (ARC-IUSP) can be solved (or proven to be infeasible) in polynomial time by a standard cutting plane algorithm.

Clearly, the optimal solution value  $\lambda_{\max}^*$  of the linear programming relaxation of (ARC-IUSP) is a lower bound for the optimal solution value of MIN-ARC-IUSP. An optimal solution  $\lambda^*$  of the linear relaxation corre-

sponds to an optimal ‘fractional’ solution of MIN-ARC-IUSP, i.e., a real-valued metric that are compatible with the given path set  $S$  and minimize the largest arc length (among all those fractional metrics with  $\lambda_a \geq 1$  for all  $a \in A$  and with  $\lambda(P') - \lambda(P) \geq 1$  for all prescribed shortest paths  $P \in S$  and all non-shortest paths  $P' \in \mathcal{P}(s_P, t_P) \setminus \{P\}$ ). Using standard scaling and rounding techniques, such as Cramer’s rule or iterative rounding, one can easily transform such a fractional solution into an integer solution of (ARC-IUSP), which then defines a feasible solution for MIN-ARC-IUSP.

The currently best known approximation ratio is achieved by algorithm MIP-Rounding shown below. This algorithm first solves the linear programming relaxation of (ARC-IUSP) and then, depending on the length of the longest path in  $S$ , simultaneously scales all fractional arc lengths and rounds the outcome to the nearest integer. A variant of this algorithm was originally proposed by Ben-Ameur and Gourdin [26, 27] for the undirected version of the MIN-ARC-IUSP problem.

**Theorem 4.9 (Ben-Ameur and Gourdin [27])** *Algorithm MIP-Rounding has a worst-case approximation ratio of  $\min(|V|/2, \max_{P \in S} |P|)$  for MIN-ARC-IUSP.*

In order to prove Theorem 4.9, we first need to analyze some redundancies among the inequalities in (ARC-IUSP). A large portion of the redundant inequalities can be characterized by the notion of weak disjointness.

**Definition 4.10** *We say that two  $(s, t)$ -paths  $P_1$  and  $P_2$  are **weakly disjoint** if there are two distinct nodes  $u$  and  $v$  that occur in this order in both paths (i.e.,  $P_1[u, v] \neq \emptyset$  and  $P_2[u, v] \neq \emptyset$ ) and*

$$(i) \ P_1[s, u] = P_2[s, u],$$

---

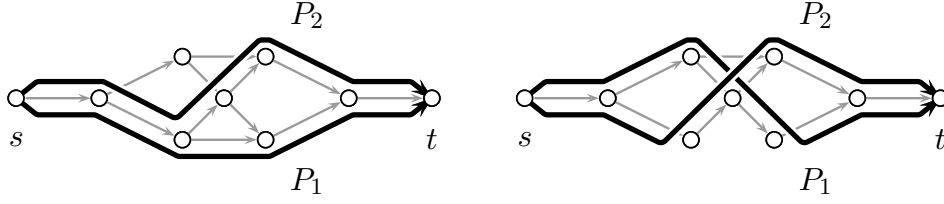
**Algorithm 4.2** MIP-Rounding

---

**Input:** Digraph  $D = (V, A)$  and path set  $S \subseteq \mathcal{P}$ .

**Output:** Integer metric  $\lambda \in \mathbb{Z}_+^a$  with  $\lambda_a \geq 1$  for all  $a \in A$  that is compatible with  $S$ .

1. Solve the linear relaxation of (ARC-IUSP) for  $S$ .
  2. If (ARC-IUSP) is not solvable, then  
return ‘ $S$  is no USPS’.
  3. Let  $(\lambda^*, \lambda_{\max}^*)$  be the optimal fractional solution of (ARC-IUSP).
  4. Let  $M := \max_{P \in S} |P|$ .
  5. If  $M < |V|/2$ , then
    - 5.1. return metric  $\lambda_a := \lceil \lambda_a^* \cdot M \rceil$  for all  $a \in A$ ,  
else
    - 5.2. return metric  $\lambda_a := \lceil \lambda_a^* \cdot |V|/2 - 1/2 \rceil$  for all  $a \in A$ .
-



**Figure 4.9** Examples for weakly disjoint paths (left) and not weakly disjoint paths (right).

(ii)  $P_1[v, t] = P_2[v, t]$ , and

(iii)  $P_1[u, v]$  and  $P_2[u, v]$  are (internally) node disjoint.

For an  $(s, t)$ -path  $P$ , the set  $\mathcal{WD}(P) \subset \mathcal{P}(s, t)$  denotes all weakly disjoint paths for  $P$ .

Two weakly disjoint paths must be disjoint except for a common subpath at the start and a common subpath at the end. These subpaths may be empty, i.e., a pair of disjoint  $(s, t)$ -paths is also weakly disjoint. Figure 4.9 illustrates the weak disjointness of paths.

**Lemma 4.11** In (ARC-IUSP), inequality (4.8) is redundant for all paths  $P \in S$  and  $P' \in \mathcal{P}(s_P, t_P) \setminus \{P\}$  that are not weakly disjoint.

**Proof.** Let  $P \in S$  and  $P' \in \mathcal{P}(s_P, t_P) \setminus \{P\}$  be not weakly disjoint. Then there exists a node  $v \in P$ , such that  $\emptyset \neq P[s_P, v] \neq P'[s_P, v] \neq \emptyset$  and  $\emptyset \neq P[v, t_P] \neq P'[v, t_P] \neq \emptyset$ . Summing inequalities (4.8) for the path pair  $P$  and  $P'[s_P, v] \oplus P[v, t_P]$  and for the path pair  $P$  and  $P[s_P, v] \oplus P'[v, t_P]$  yields

$$\sum_{a \in P'} \lambda_a - \sum_{a \in P} \lambda_a \geq 2,$$

which dominates inequality (4.8) for the paths  $P$  and  $P'$ .  $\square$

By Lemma 4.11, the integer linear program (ARC-IUSP) is equivalent to

$$\begin{aligned} & \min \lambda_{\max} && (\text{ARC-IUSLP}') \\ & \sum_{a \in P'} \lambda_a - \sum_{a \in P} \lambda_a \geq 1 && \forall P \in S, P' \in \mathcal{WD}(P), \\ & 1 \leq \lambda_a \leq \lambda_{\max} && \forall a \in A, \text{ and} \\ & \lambda_a \in \mathbb{Z}_+ && \forall a \in A, \end{aligned} \quad (4.9)$$

and the linear relaxation (ARC-IUSP) is equivalent to the linear relaxation (ARC-IUSLP'). In order to prove Theorem 4.9, it therefore is sufficient to

show that algorithm **MIP-Rounding** rounds any optimal fractional solution of (ARC-IUSLP') to an integer solution of (ARC-IUSLP').

**Proof [Theorem 4.9].** Consider an MIN-ARC-IUSP instance given by the digraph  $D$  and the path set  $S$ . If  $S$  is not a USPS, then the linear relaxation of (ARC-IUSLP') has no solution and the algorithm returns infeasibility. So, assume  $S$  is a USPS and denote by  $(\lambda^*, \lambda_{\max}^*)$  the optimal fractional solution of (ARC-IUSLP'). Clearly,  $\lceil \lambda_{\max}^* \rceil$  is a lower bound for the optimal integer solution value.

First, consider the case where  $M := \max_{P \in S} |P| < |V|/2$ . Let  $\bar{\lambda}_a := \lambda_a^* \cdot M$ , for all  $a \in A$ . **MIP-Rounding** returns arc lengths  $\lambda_a = \lceil \bar{\lambda}_a \rceil$ , for all  $a \in A$ . Because  $\lambda^*$  satisfies all inequalities (4.9), we have  $\bar{\lambda}(P') - \bar{\lambda}(P) \geq M$  for all  $P \in S$  and  $P' \in \mathcal{WD}(P)$ . With  $\bar{\lambda}_a \leq \lambda_a < \bar{\lambda}_a + 1$ , it follows that  $\lambda(P') - \lambda(P) > 0$  for all  $P \in S$  and  $P' \in \mathcal{WD}(P)$ . This implies that all paths in  $S$  are indeed unique shortest paths w.r.t.  $\lambda$ . Hence,  $\lambda$  is a compatible arc length function for  $S$  with

$$\lambda_{\max} = \max_{a \in A} \lambda_a \leq \lceil M \cdot \lambda_{\max}^* \rceil \leq M \cdot \lceil \lambda_{\max}^* \rceil.$$

Now, consider the case where  $M \geq |V|/2$ . Let  $\bar{\lambda}_a := \lambda_a^* \cdot |V|/2$ , for all  $a \in A$ . Analogous to the first case, we observe that  $\bar{\lambda}(P') - \bar{\lambda}(P) \geq |V|/2$  for all  $P \in S$  and  $P' \in \mathcal{WD}(P)$ . Because  $P$  and  $P'$  are weakly disjoint, there are at most  $|V|$  many arcs in  $P \cup P'$ . With  $\bar{\lambda}_a - 1/2 \leq \lambda_a < \bar{\lambda}_a + 1/2$ , we therefore have  $\lambda(P') - \lambda(P) > 0$  for all  $P \in S$  and  $P' \in \mathcal{WD}(P)$ . Hence,  $\lambda$  is compatible with  $S$  and

$$\lambda_{\max} = \max_{a \in A} \lambda_a \leq \left\lceil \frac{|V|}{2} \cdot \lambda_{\max}^* - \frac{1}{2} \right\rceil \leq \frac{|V|}{2} \cdot \lceil \lambda_{\max}^* \rceil.$$

This concludes the proof.  $\square$

Analogously, we can compute approximate solutions for the MIN-PATH-IUSP problem. The corresponding integer linear programming formulation for MIN-PATH-IUSP is

$$\begin{aligned} \min \quad & \lambda_{\max} && (\text{PATH-IUSP}) \\ \sum_{a \in P'} \lambda_a - \sum_{a \in P} \lambda_a \geq 1 && \forall P \in S, P' \in \mathcal{P}(s_P, t_P) \setminus \{P\} \\ \sum_{a \in P} \lambda_a \leq \lambda_{\max} && \forall P \in S \\ \lambda_a \geq 1 && \forall a \in A \\ \lambda_a \in \mathbb{Z}_+ && \forall a \in A. \end{aligned}$$

Applying algorithm **MIP-Rounding** to the linear relaxation of (PATH-IUSP), we obtain the same worst-case approximation guarantee for MIN-PATH-IUSP as for MIN-ARC-IUSP.

**Theorem 4.12** *Algorithm MIP-Rounding achieves a worst-case approximation ratio of  $\min(|V|/2, \max_{P \in S} |P|)$  for MIN-PATH-IUSP, if applied to the linear relaxation of (PATH-IUSP).*

**Proof.** Analogous to Theorem 4.9.  $\square$

The same approximation ratio is also achieved for the corresponding undirected problem versions.

**Remark 4.13** The approximation algorithm MIP-Rounding presented in this section are mainly of theoretical interest. For instances arising in real-world telecommunication network planning, the integer linear programming models for MIN-ARC-IUSP and MIN-PATH-IUSP usually can be solved to optimality very efficiently with state-of-the-art integer linear programming solvers.

## 4.6 Unique Shortest Path Forwardings

In this final section we consider an alternative representation of unsplittable shortest path routings. Instead of the end-to-end routing paths themselves, we are now given a set of so-called forwarding arcs.

**Definition 4.14** A **forwarding arc** is a pair  $(t, (u, v))$  of a destination node  $t \in V$  and an arc  $(u, v) \in A$ , and a given forwarding arc  $(t, (u, v))$  means that arc  $(u, v)$  is the only arc emanating from node  $u$  that is contained in any shortest  $(u, t)$ -path. A **forwarding** is a set  $F \subseteq V \times A$  of forwarding arcs.

The forwarding representation of unsplittable shortest path routings corresponds to the representation that is used internally by the routers in a real network. Recall that each router maintains a local forwarding table describing via which arc it has to forward the packets to which destination. For an unsplittable shortest path routing or, more general, for an unsplittable source invariant routing, this table assigns exactly one emanating link to each possible destination.

Prescribing a forwarding  $F \subset V \times A$  corresponds to prescribing some of the forwarding table entries at the routers. A given forwarding arc  $(t, (u, v)) \in F$  forces the forwarding table of router  $u$  to map destination  $t$  to the emanating link  $(u, v)$  and thereby ensures that all traffic from or via router  $u$  to destination  $t$  is sent via this link.

Note that a forwarding not necessarily prescribes entire end-to-end paths. A single forwarding arc  $(t, (u, v))$  with  $t \neq v$ , for example, only enforces that the  $(u, t)$ -routing path(s) leave node  $u$  via arc  $(u, v)$ , but it does not define how the path(s) continue after node  $v$  on.

As the forwarding representation allows this partial prescription of end-to-end paths, it is very useful in optimization models for unsplittable shortest path routings. Several types of conflicts in a given non-unique shortest path routing (i.e., substructures proving that the given routing is no unsplittable shortest path routing) can be recognized already when only some forwarding arcs have been determined. Optimization methods based on the forwarding representation thus can recognize these types of infeasibilities more efficiently than methods based on the end-to-end path representation. Of course, the forwarding representation is also very handy if the routing is given in terms of the routers' forwarding tables anyway.

In this thesis, we are interested in those forwardings that correspond or can be extended to unsplittable shortest path routings. Analogous to path sets, we therefore introduce the notion of a compatible metric and a unique shortest path forwarding.

**Definition 4.15**

- (i) A metric  $\lambda \in \mathbb{R}_+^A$  is said to be **compatible** with a forwarding  $F \subseteq V \times A$  if, for each  $(t, (u, v)) \in F$ , arc  $(u, v)$  is contained in all shortest  $(u, t)$ -paths with respect to  $\lambda$ .
- (ii) A forwarding  $F \subseteq V \times A$  is called a **unique shortest path forwarding (USPF)** if there exists a compatible metric for  $F$ . Otherwise  $F$  is called a **non-USPF**.

Note that, according to this definition, a metric  $\lambda$  that is compatible with a forwarding  $F$  not necessarily induces unique shortest paths, not even for those node pairs  $(u, t)$  for which  $F$  contains a forwarding arc  $(t, (u, v))$ . Definition 4.15 only enforces that the given forwarding arcs  $(t, (u, v))$  are uniquely determined for every compatible metric, that is, *all* shortest  $(u, t)$ -paths contain arc  $(u, v)$ . Yet, if the given forwarding corresponds to a set of complete end-to-end paths, then each metric that is compatible with the forwarding also uniquely induces these end-to-end paths.

**Definition 4.16** For each path set  $S \subseteq \mathcal{P}$  its **associated forwarding** is

$$\begin{aligned} \mathcal{F}(S) &:= \bigcup_{P \in S} \{(t_P, a) : a \in P\} \\ &= \{(t, a) \in V \times A : a \in P \text{ for some } (s, t)\text{-path } P \in S\}. \end{aligned}$$

**Observation 4.17** A metric is compatible with a path set  $S \subseteq \mathcal{P}$  if and only if it is compatible with its associated forwarding  $\mathcal{F}(S)$ .

Observation 4.17 implies that a path set  $S \subseteq \mathcal{P}$  is a USPS if and only if its associated forwarding  $\mathcal{F}(S)$  is a USPF. Note, however, that if  $S$  is a USPS

and  $F \subset \mathcal{F}(S)$  but  $F \neq \mathcal{F}(S)$ , then a metric that is compatible with  $S$  is not necessarily also compatible with  $F$ .

The task of finding a compatible metric for a given forwarding  $F$  and, thereby, deciding whether  $F$  is a USPF is formalized as the INVERSE UNIQUE SHORTEST PATH FORWARDING problem:

<i>Problem:</i>	IUSPF
<i>Instance:</i>	A digraph $D = (V, A)$ and a forwarding $F \subseteq V \times A$ .
<i>Task:</i>	Find a metric $\lambda \in \mathbb{R}_+^A$ that is compatible with $F$ or prove that none exists.

The IUSPF problem is closely related to the IUSP problem. By Observation 4.17, a IUSP instance consisting of a digraph  $D$  and a path set  $S$  is equivalent to the IUSPF instance consisting of  $D$  and the associated forwarding  $\mathcal{F}(S)$ .

Like the IUSP problem, also the IUSPF problem can be solved using linear programming techniques. In the following, we present a formulation of IUSPF as a linear inequality system with only polynomially many variables and inequalities. Variants of this formulation have been used by Ben-Ameur and Gourdin [27], Bley and Koch [33, 36], and Prytz [169] to find compatible metrics for both unsplittable shortest path systems and unsplittable shortest path forwardings. Broström and Holmberg [44, 45, 46] used a modification of this formulation for finding metrics that induce a prescribed shortest multi-path routing.

We begin by reviewing some well known facts about shortest paths and shortest path (anti-)arborescences in digraphs. For notational simplicity we assume that the digraph  $D$  is simple and strongly connected in the remainder of this section.

**Definition 4.18** *Let  $D = (V, A)$  be a simple digraph and let  $\lambda \in \mathbb{R}_+^A$ .*

(i) *A **potential**  $\pi = (\pi_v)_{v \in V} \in \mathbb{R}^V$  is called **feasible** if*

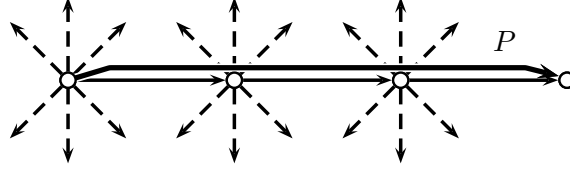
$$\lambda_{(u,v)} + \pi_u - \pi_v \geq 0 \quad \text{for all } (u, v) \in A. \quad (4.10)$$

(ii) *An arc  $(u, v) \in A$  is called **tight** for the metric  $\lambda$  and the feasible potential  $\pi$  if (4.10) holds with equality for  $(u, v)$ .*

It is well known that an  $(s, t)$ -path  $P$  is a (not necessarily unique) shortest  $(s, t)$ -path with respect to  $\lambda$  if and only if there exist a feasible potential  $\pi$  such that all arcs in  $P$  are tight, see Bellman [21], Dijkstra [73], or Tarjan [183], for example.

A characterization of the metric for which an arc  $(u, v)$  is contained in all shortest  $(u, t)$ -paths is given by the following proposition.





**Figure 4.10** Arc set  $\bar{P}$  for a path  $P$ : Arcs in  $P$  are solid, arcs in  $\bar{P}$  are dashed.

**Proposition 4.19** *An arc  $(u, v)$  is contained in all shortest  $(u, t)$ -paths with respect to  $\lambda$  if and only if there exist a feasible potential  $\pi$  such that  $(u, v)$  is tight and all arcs in  $\delta^+(u) \setminus \{(u, v)\}$  are not tight with respect to  $\pi$ .*

**Proof.** Let  $\lambda \in \mathbb{R}_+^A$ . First, we show that there exists a feasible potential with the required properties if  $(u, v)$  is contained in all shortest  $(u, t)$ -paths.

For each node  $v \in V$ , let  $\pi_v$  be the negative length of a shortest  $(v, t)$ -path, i.e.,  $\pi_v := -\min\{\lambda(P') : P' \in \mathcal{P}(v, t)\}$ . One easily verifies that  $\pi$  is a feasible potential.

For any shortest  $(u, t)$ -path  $P$ , all arcs  $a \in P$  are tight. Since it is contained in all shortest  $(u, t)$ -paths, arc  $(u, v)$  is tight.

Now suppose one of the arcs  $(u, w) \in \delta^+ \setminus \{(u, v)\}$  was tight. Let  $P_{w,t}^*$  be some shortest  $(w, t)$ -path with respect to  $\lambda$ . As all arcs in  $P_{w,t}^*$  are tight as well, it follows that the path  $(u, w) \oplus P_{w,t}^*$  is a shortest  $(u, t)$ -path. This, however, contradicts with the assumption that *all* shortest  $(u, t)$ -path contain the arc  $(u, v)$ . Hence, the potential  $\pi$  constructed above has the required properties.

Next, we show that there exists no feasible potential with the required properties if  $(u, v)$  is not contained in all shortest  $(u, t)$ -paths. So, suppose that there are two arcs  $(u, v)$  and  $(u, w)$  that are contained in the shortest  $(u, t)$ -paths  $P_1^*$  and  $P_2^*$ , respectively. Since both  $P_1^*$  and  $P_2^*$  are shortest  $(u, t)$ -paths, all arcs that are contained in either path must be tight with respect to any feasible potential  $\pi$  for  $\lambda$ . In particular, both arcs  $(u, v)$  and  $(u, w)$  must be tight. Hence, there exists no feasible potential  $\pi$  with respect to  $\lambda$  such that  $(u, v)$  is tight but  $(u, w) \in \delta^+ \setminus \{(u, v)\}$  is not.  $\square$

The uniqueness of a shortest path  $P$  with respect to a metric  $\lambda$  can be characterized by applying Proposition 4.19 iteratively to all arcs of  $P$ . Let  $P$  be an  $(s, t)$ -path and let  $\bar{P} \subset A$  be the set of all those arcs that emanate from the nodes  $\{u \in P : u \neq t\}$  but that are not contained in  $P$ , i.e.,

$$\bar{P} := \bigcup_{(u,v) \in P} (\delta^+(u) \setminus \{(u, v)\}) .$$

An illustration of the set  $\bar{P}$  is given in Figure 4.10.

**Proposition 4.20** *The path  $P$  is the unique shortest  $(s, t)$ -path with respect to  $\lambda$  if and only if there exist a feasible potential  $\pi$  such that all arcs in  $P$*

are tight and all arcs in  $\bar{P}$  are not tight with respect to  $\pi$ .

**Proof.** Follows immediately from Proposition 4.19.  $\square$

More important in our context, Proposition 4.19 leads straightforward to a characterization of all those metrics that are compatible with a given forwarding. Let  $F \subseteq V \times A$  be the given forwarding and denote

$$\bar{F} := \bigcup_{(t, (u, v)) \in F} \{ (t, a) : a \in \delta^+(u) \setminus \{(u, v)\} \} .$$

**Proposition 4.21** *A metric  $\lambda \in \mathbb{R}_+^A$  is compatible with a forwarding  $F \subseteq V \times A$  if and only if there exists a feasible potential  $\pi^t = (\pi_v^t)_{v \in V} \in \mathbb{R}_+^V$  for each node  $t \in V$  such that*

- (i) arc  $(u, v)$  is tight with respect to  $\pi^t$  and  $\lambda$  for each  $(t, (u, v)) \in F$ , and
- (ii) arc  $(u, v)$  is non-tight with respect to  $\pi^t$  and  $\lambda$  for each  $(t, (u, v)) \in \bar{F}$ .

**Proof.** Follows directly from Proposition 4.19.  $\square$

Proposition 4.21 provides a complete characterization of all those metrics (and the corresponding feasible potentials) that are compatible with a given forwarding.

This characterization, however, uses the non-tightness of arcs, which corresponds to strict inequalities to be satisfied. In order to obtain a standard linear programming formulation with only non-strict linear inequalities for the IUSPF problem, we apply the same scaling argument as in Section 4.3 for the path case.

Clearly, the potential  $\pi$  is a feasible with respect to the metric  $\lambda$  if and only if  $c\pi$  is a feasible with respect to  $c\lambda$ , for any  $c > 0$ . Furthermore, an arc  $a$  is tight with respect to  $\lambda$  and  $\pi$  if and only if it is tight with respect to  $c\lambda$  and  $c\pi$ . Hence, there exists a factor  $c > 0$  such that after scaling the metric  $\lambda$  and the feasible potential  $\pi$  by  $c$  each tight inequality (4.10) remains tight and the slack of each non-tight inequality (4.10) is at least 1. (With the same arguments as in Proposition 4.3, we can even ensure that the metric  $\lambda$  and the feasible potential  $\pi$  are integer-valued.)

Together with Proposition 4.21, it follows that the IUSPF problem is equivalent to the problem of finding a solution of the following linear inequality system:

$$\lambda_{(u, v)} + \pi_u^t - \pi_v^t = 0 \quad \forall (t, (u, v)) \in F \quad (4.11)$$

$$\lambda_{(u, v)} + \pi_u^t - \pi_v^t \geq 1 \quad \forall (t, (u, v)) \in \bar{F} \quad (4.12)$$

$$\lambda_{(u, v)} + \pi_u^t - \pi_v^t \geq 0 \quad \forall (t, (u, v)) \in (V \times A) \setminus F \setminus \bar{F} \quad (4.13)$$

$$\lambda_a \geq 0 \quad \forall a \in A \quad (4.14)$$

The variables  $\lambda_a$ ,  $a \in A$ , describe the metric we wish to find and, for each  $t \in V$ , the variables  $\pi_s^t \in \mathbb{R}$ ,  $s \in V$ , describe a feasible potential for the shortest paths towards node  $t$ .

**Theorem 4.22** *A forwarding  $F \subseteq V \times A$  is a USPF if and only if the linear system (4.11)–(4.14) has a solution. Furthermore, each solution  $(\lambda, \pi)$  of (4.11)–(4.14) defines a compatible metric  $\lambda$  for  $F$ .*

**Proof.** Follows straightforward from Proposition 4.21.  $\square$

The number of variables and linear constraints in the system (4.11)–(4.14) is linear in  $|V| \cdot |A|$ . With the ellipsoid method [103], for example, (4.11)–(4.14) thus can be solved in polynomial time.

**Theorem 4.23** *The IUSPF problem is solvable in polynomial time.*

**Remark 4.24** Note that the given forwarding  $F$  is not required to have the Bellman property or any other of the properties discussed in Chapter 5. In fact, we do not even require that  $F$  and  $\bar{F}$  are disjoint. In principle, the linear system (4.11)–(4.14) may be formulated for *any* forwarding  $F \subseteq V \times A$ . Of course, (4.11)–(4.14) is infeasible if the sets  $F$  and  $\bar{F}$  are not disjoint, because it then contains the conflicting constraints  $\lambda_{(u,v)} + \pi_u^t - \pi_v^t = 0$  and  $\lambda_{(u,v)} + \pi_u^t - \pi_v^t \geq 1$  for any  $(t, (u, v)) \in F \cap \bar{F}$ .

Together with Observation 4.17, Theorems 4.22 and 4.23 also yield an alternative, polynomial time solvable linear programming formulation for the IUSP problem discussed in Section 4.3.

**Corollary 4.25** *A path set  $S \subseteq \mathcal{P}$  is a USPS if and only if the linear system (4.11)–(4.14) has a solution for its associated forwarding  $\mathcal{F}(S)$ . Furthermore, each solution  $(\lambda, \pi)$  of (4.11)–(4.14) defines a compatible metric  $\lambda$  for  $S$ .*

**Corollary 4.26** *The IUSP problem is solvable in polynomial time.*

**Remark 4.27** The fact that (4.11)–(4.14) contains only polynomially many variables and constraints might suggest that it is better suited for solving the IUSP problem than the exponentially large formulation (4.3)–(4.4) discussed in Section 4.3. In our computational experiments with real-world IUSP problems, however, solving the compact linear system (4.11)–(4.14) usually was more time consuming than solving the exponentially large system (4.3)–(4.4) by a cutting plane algorithm that iteratively separates violated inequalities of type (4.3).

As for the path case, one might be interested in optimization versions of the INVERSE UNIQUE SHORTEST PATH FORWARDING problem, where the task is to find a metric of strictly positive integer arc lengths that are compatible with a given forwarding and that minimize an objective function such

as the maximum arc length, the maximum implied distance between two nodes, or the sum of all arc lengths, for example. With Observation 4.17, the  $9/8 - \epsilon$  inapproximability threshold for MIN-ARC-IUSP carries over immediately to the problem MIN-ARC-IUSPF of finding a metric that minimizes the largest arc length. MIN-ARC-IUSPF also can be approximated easily within a factor of  $|V|$  applying a variation of the scaling rounding approach of MIP-Rounding to the optimal solution of linear relaxation of the following integer programming formulation of MIN-ARC-IUSPF:

$$\begin{array}{ll}
 \min \lambda_{\max} & \text{(ARC-IUSPF)} \\
 \lambda_{(u,v)} + \pi_u^t - \pi_v^t = 0 & \forall (t, (u, v)) \in F \\
 \lambda_{(u,v)} + \pi_u^t - \pi_v^t \geq 1 & \forall (t, (u, v)) \in \bar{F} \\
 \lambda_{(u,v)} + \pi_u^t - \pi_v^t \geq 0 & \forall (t, (u, v)) \in (V \times A) \setminus F \setminus \bar{F} \\
 1 \leq \lambda_a \leq \lambda_{\max} & \forall a \in A \\
 \lambda_a \in \mathbb{Z}_+ & \forall a \in A
 \end{array}$$

From the practical point of view, however, these optimization versions of the IUSPF problem are not very important. For forwardings that correspond to complete end-to-end path sets, they are equivalent to the corresponding inverse unique shortest path problem versions. Forwardings that do not correspond to end-to-end path sets, on the other hand, only arise in intermediate steps of the network and routing planning process, where the integrality and the boundedness of the routing lengths is of no concern.

## 4.7 Concluding Remarks

Another problem variant MIN-SUM-IUSP, where the objective is to minimize the sum of all arc lengths, can be shown to be  $\mathcal{APX}$ -hard with an approximation preserving reduction from the MINIMUM VERTEX COVER problem. Because of the lesser practical relevance of the MIN-SUM-IUSP problem we omit this proof here.

**Theorem 4.28** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate MIN-SUM-IUSP within a factor of  $91/90 - \epsilon$ .*

The same inapproximability bound also holds for the corresponding MIN-SUM-IUSPF problem, where a forwarding instead of a path set is given.

It remains open whether there are constant factor approximation algorithms for MIN-PATH-IUSP, MIN-ARC-IUSP, MIN-SUM-IUSP, or the corresponding IUSPF problem versions. The currently best known algorithms are based on linear programming and rounding and achieve a worst-case approximation ratio of  $\mathcal{O}(|V|)$  for all three objective functions.

---

In this chapter, we discussed only the case of unique shortest paths and unique shortest path forwarding. All problems can be defined analogously for the case where multiple shortest  $(s, t)$ -paths or multiple arcs emanating from  $s$  to destination  $t$  may be prescribed for some or all node pairs  $(s, t) \in V \times V$  and we seek for a metric such that exactly the prescribed paths or forwarding arcs are induced, but no more parallel paths or forwarding arcs. These problem variants arise in the planning of OSPF networks with traffic splitting among equally long shortest paths, so-called ECMP routing, for example. The results presented in this chapter carry over straightforward to these problem variants.



## Chapter 5

# Unique Shortest Path Systems

One of the major difficulties with unsplittable shortest path routing is that the different commodities' end-to-end routing paths cannot be chosen independent of each other. The unsplittable shortest path routing paradigm enforces many rather subtle and complicated interdependencies among the paths that comprise a valid routing. Given a digraph  $D = (V, A)$ , we say that a path set  $S \subseteq \mathcal{P}$  is a unique shortest path systems (USPS) if there exist a compatible metric  $\lambda \in \mathbb{R}_+^A$  for  $S$ , i.e., a metric  $\lambda$  such that each path  $P \in S$  is the unique shortest path between its terminals. The family of USPSs corresponds to exactly those routings that are realizable as an unsplittable shortest path routing when ignoring the limited range of the admissible routing lengths in real shortest path routing protocols.

In this chapter, we discuss the combinatorial properties of these unique shortest path sets and study the problems of finding a maximal USPS or a minimal non-USPS within a given path set.

After reviewing the relevant literature in Section 5.1, we introduce some basic notions and notations related to unique shortest path systems in Section 5.2. In Section 5.3, we examine the Bellman property and some other properties of unique shortest path systems. These properties can be regarded as necessary conditions for a path set to be a USPS, but they are not sufficient to completely characterize the family of USPSs in a general digraph.

In Section 5.4 is dedicated to the problem of finding an irreducible non-USPS within a given path set. An irreducible non-USPS is an inclusion-wise minimal set of conflicting paths, and finding these minimal conflicts in a given path set provides useful information on how to modify the paths in order to obtain a valid unsplittable shortest path routing. We review the standard greedy algorithm for computing such non-USPSs and present some simple but effective improvements of this algorithm. We also show that the problem of finding a minimum cardinality or minimum weight such

non-USPS is  $\mathcal{NP}$ -hard to approximate within a factor less than  $7/6$ .

The reverse problem of finding an inclusion-wise maximal USPS in a given path set is addressed in Section 5.5. Again, we review the standard greedy algorithm for solving this problem and prove that the problem of finding a maximum cardinality or maximum weight such set is computationally hard.

In Section 5.6, we consider unique shortest path systems in undirected graphs and review a special class of undirected graphs where the Bellman property alone completely characterizes the family all USPSs.

Section 5.7 finally is dedicated to unique shortest path forwardings, an alternative representation of unsplittable shortest path routings. Similar to the forwarding tables used by the routers in practice, a forwarding describes via which arcs the traffic streams leave the different nodes on the ways to their respective destinations. Most combinatorial properties, complexity results, and algorithms carry over straightforward from unique shortest path systems to unique shortest path forwardings (USPF). The problem of finding a minimal non-USPF that is contained in a given forwarding is discussed in detail, because it is relevant in later chapters of this thesis. As for the path-set variant of the problem, we present a polynomial time greedy-like algorithm for finding an inclusion-wise minimal such non-USPF, and we show that the problem of finding a minimum cardinality or minimum weight such non-USPF is  $\mathcal{NP}$ -hard to approximate within a factor strictly less than  $7/6$ .

## 5.1 Related Work

The Bellman or sub-optimality property was first observed by Bellman et al. [21, 23, 24], Dijkstra [73], and Tarjan [183] in the context of shortest path algorithms and dynamic programming: Every subpath of a shortest path is a shortest path as well. As a fundamental property of unique shortest path systems it was discussed later by Ben-Ameur and Gourdin [27], Bley and Koch [36], Staehle et al. [180], Milbrandt [141], and Prytz [169], often referred to as *sub-path property* or *sub-optimality of paths*. Fortz and Thorup [92], Broström and Holmberg [45, 46], de Giovanni et al. [69], and Tomaszewski et al. [184, 197] considered a multi-path variant of the Bellman property for path sets that arise from multi-shortest path routing with equal traffic splitting.

Ben-Ameur and Gourdin [27] introduced the notion of a compatible metric and discussed the Bellman property, the cyclic comparability property, and the generalized cyclic comparability property for undirected unique shortest path systems. They also proposed several linear and integer linear programming formulations for finding a compatible metric for a given path set, which we discussed in the previous chapter.



Broström and Holmberg [44, 45, 46] derived a further property of unique shortest path systems by analyzing a special case of infeasibilities that may arise in the linear programming formulation of the corresponding inverse (unique) shortest paths problem. This property corresponds to the non-existence of an augmenting cycle in some related multicommodity flow problem, it is stronger than the Bellman property, and it still can be verified for a given path set in polynomial time.

The problem of finding an inclusion-wise minimal conflict in a given routing was first considered by Bley and Koch [33, 36], who computed small (but not necessarily irreducible) conflicts within a given candidate shortest path routing from irreducible inconsistent subsystems of the corresponding inverse unique shortest path problem's LP. Bley and Koch used the found conflicts to generate violated inequalities in a Branch-and-Cut algorithm for some (survivable) network design problem with unique shortest path routing. Prytz [169] generalized this approach to solve an unsplittable shortest path routing problem with multicast commodities.

## 5.2 Basic Definitions and Observations

Again, let  $D = (V, A)$  be a directed graph and  $\mathcal{P}$  be the set of all simple paths in  $D$ . As in the previous chapter, we denote the set of all  $(s, t)$ -paths between two nodes  $s$  and  $t$  by  $\mathcal{P}(s, t)$  and the source and the target of a given path  $P$  by  $s_P$  and by  $t_P$ , respectively.

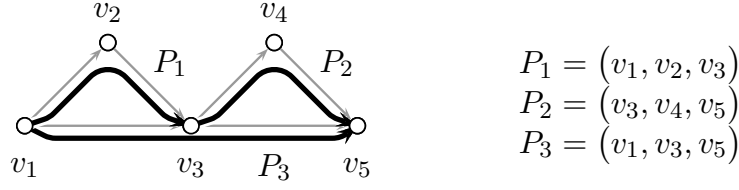
Throughout this chapter, we are interested in the combinatorial properties of unique shortest path systems. Recall that

- (i) a metric  $\lambda = (\lambda_a)_{a \in A} \in \mathbb{R}_+^A$  is said to be *compatible* with a given path set  $S \subseteq \mathcal{P}$ , if each path  $P \in S$  is the unique shortest  $(s_P, t_P)$ -path with respect to  $\lambda$ , and that
- (ii) a path set  $S \subseteq \mathcal{P}$  is called a *unique shortest path system (USPS)* if there exists a compatible metric  $\lambda \in \mathbb{R}_+^A$  for  $S$ . Otherwise  $S$  is called a *non-USPS*.

The family of USPSs corresponds to exactly those routings that are realizable as unsplittable shortest path routings if ignoring the limited range of admissible routing lengths in real shortest path routing protocols.

In the previous chapter, we have already shown that the INVERSE UNIQUE SHORTEST PATHS problem of finding a compatible metric for a given path set  $S \subseteq \mathcal{P}$  or proving that no such metric exists can be solved in polynomial time. Thus, we can decide in polynomial time whether a given path set  $S$  is a USPS or not.

It is a trivial observation that any subset of a USPS is a USPS as well: Every metric  $\lambda \in \mathbb{R}_+^A$  that is compatible with some path set  $S$  is also compatible with any subset  $R \subset S$  of these paths. Thus, the family of all USPSs



**Figure 5.1** Example showing that  $\mathcal{I}_{USPS}$  is not a matroid.

in a digraph  $D$  forms an independence system (or hereditary family). In the terminology of independence systems, each USPS is an independent set and each non-USPS is a dependent set. Accordingly, we denote the family of all USPSs by  $\mathcal{I}_{USPS}$ , i.e.,

$$\mathcal{I}_{USPS} := \{S \subseteq \mathcal{P} : S \text{ is a USPS}\} \subseteq 2^{\mathcal{P}}.$$

The following example shows that this independence system is not a matroid in general.

**Example 5.1** Consider the digraph  $D = (V, A)$  with  $V = \{v_1, \dots, v_5\}$  and  $A = \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5), (v_1, v_3), (v_3, v_5)\}$ , and the three paths  $P_1 = (v_1, v_2, v_3)$ ,  $P_2 = (v_3, v_4, v_5)$ , and  $P_3 = (v_1, v_3, v_5)$  shown in Figure 5.1.

Obviously, each of the two path sets  $S_1 := \{P_1, P_2\}$  and  $S_2 := \{P_3\}$  is a USPS. If  $\mathcal{I}_{USPS}$  were a matroid, then  $\{P_1, P_3\}$  or  $\{P_2, P_3\}$  would have to be USPSs, too. Yet,  $P_1$  and  $P_3$  contain different  $(v_1, v_3)$ -(sub)paths. As these cannot both be unique shortest paths for the same metric,  $\{P_1, P_3\}$  is not a USPS. For the same reason,  $\{P_2, P_3\}$  is no USPS either.  $\triangle$

As for any independence system, the maximal independent and minimal dependent sets (with respect to set-inclusion) are of special interest.

### Definition 5.1

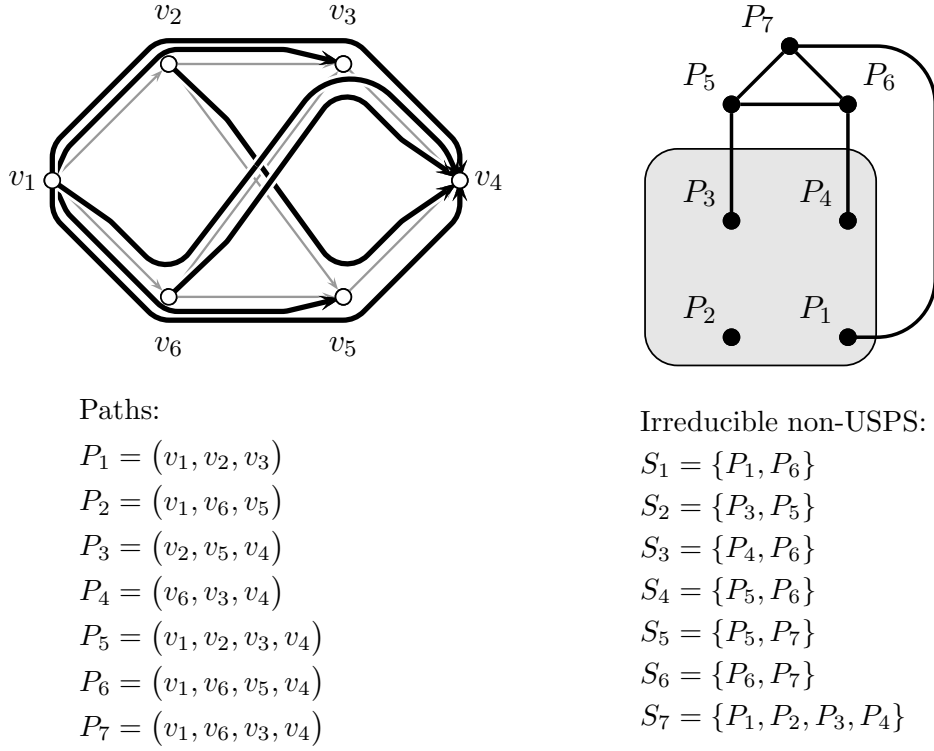
- (i) A path set  $S \subseteq \mathcal{P}$  is a **maximal USPS** if  $S \in \mathcal{I}_{USPS}$  and if there exist no USPS  $R \in \mathcal{I}_{USPS}$  with  $S \subset R$ ,  $S \neq R$ .
- (ii) A path set  $S \subseteq \mathcal{P}$  is an **irreducible non-USPS** if  $S \notin \mathcal{I}_{USPS}$  and  $R \in \mathcal{I}_{USPS}$  for every proper subset  $R \subset S$ ,  $R \neq S$ .

The maximal USPSs are the bases of the independence system  $(\mathcal{P}, \mathcal{I}_{USPS})$  and the irreducible non-USPSs are its circuits. We denote the family of all irreducible non-USPSs by  $\mathcal{C}_{USPS}$ , i.e.,

$$\mathcal{C}_{USPS} := \{S \subseteq \mathcal{P} : S \text{ is an irreducible non-USPS}\} \subseteq 2^{\mathcal{P}}.$$

As no irreducible non-USPS contains another, the family  $\mathcal{C}_{USPS}$  forms a so called clutter.

Sometimes it is helpful to think of the independence system of unsplittable shortest path routings in its hypergraph representation.



**Figure 5.2** Path set (left) and associated conflict hypergraph (right).

**Definition 5.2** The hypergraph  $\mathcal{H}_{USPS} := (\mathcal{P}, \mathcal{C}_{USPS})$  is called the **conflict hypergraph** of unique shortest path systems in  $D$ .

The conflict hypergraph  $\mathcal{H}_{USPS}$  is just the circuit hypergraph associated with the independence system  $(\mathcal{P}, \mathcal{I}_{USPS})$ . It contains a vertex for every path  $P \in \mathcal{P}$  and a hyperedge for every irreducible non-USPS  $S \in \mathcal{C}_{USPS}$ . Figure 5.2 shows an example of a digraph with some paths and the corresponding subgraph of the associated conflict hypergraph. The (weakly) stable sets in the conflict hypergraph  $\mathcal{H}_{USPS}$  are exactly the unique shortest path systems in  $D$ . (A subset of nodes in a hypergraph is called (weakly) stable if it contains no hyperedge.) The stable set problem in graphs and hypergraphs is well-understood and various practically efficient algorithms for solving maximum (weight) stable set problems are available. Via the hypergraph representation of USPSs, many of these results and algorithms carry over straightforward to the corresponding problems on unique shortest path systems.

From the independence system representation of unique shortest path systems we inherit a straightforward notion of the rank of a path set.

**Definition 5.3** The **rank**  $r(S)$  of a path set  $S \subseteq \mathcal{P}$  (with respect to the independence system of USPSs) is the cardinality of the largest USPS  $R$  that is fully contained in  $S$ , i.e.,

$$r(S) := \max\{|R| : R \subseteq S \text{ and } R \in \mathcal{I}_{USPS}\}.$$

As the paths in a USPS are *unique* shortest paths with respect to some metric, any USPS contains at most one  $(s, t)$ -path for each node pair  $(s, t) \in V \times V$ . With

$$K_{\max} := \{(s, t) \in V \times V : \mathcal{P}(s, t) \neq \emptyset\} ,$$

this implies that

$$|S| \leq |K_{\max}| \quad \text{for each USPS } S \in \mathcal{I}_{USPS}. \quad (5.1)$$

Clearly, (5.1) holds with equality if and only if  $S$  contains an  $(s, t)$ -path for every pair of nodes  $s, t$  that are connected in  $D$ . It is not difficult to see that any USPS with less paths can be extended to a USPS of size  $|K_{\max}|$  by adding appropriate paths for missing  $(s, t)$  pairs.

**Proposition 5.4** *Let  $S \subseteq \mathcal{P}$  be a USPS with  $|S| < |K_{\max}|$ . Then there exists a USPS  $R$  with  $S \subset R$  and  $|R| = |K_{\max}|$ .*

**Proof.** Let  $\lambda \in \mathbb{Z}_+^A$  be an integer-valued metric that is compatible with  $S$  and satisfies  $\lambda_a \geq 1$  for all  $a \in A$ . By Proposition 4.3 on page 44, such a metric exists for any USPS. Furthermore, let  $\text{idx} : A \leftrightarrow \{0, \dots, |A| - 1\}$  be an arbitrary bijection. We define a new integer-valued metric  $\lambda' \in \mathbb{Z}_+^A$  by scaling and perturbing the original metric  $\lambda$  in such a way, that *all* shortest paths are unique. This can be easily achieved by setting

$$\lambda'_a := 2^{|A|} \lambda_a + 2^{\text{idx}(a)} \quad \text{for all } a \in A.$$

One easily observes that  $\lambda'(P) < \lambda'(P')$  for all paths  $P, P' \in \mathcal{P}$  with  $\lambda(P) < \lambda(P')$ . Thus, the scaled and perturbed metric  $\lambda'$  is still compatible with  $S$ . On the other hand, the binary perturbation breaks all ties that might have existed with respect to the original metric  $\lambda$ . More precisely, we have  $\lambda'(P) \neq \lambda'(P')$  for *all* pairs of paths  $P, P' \in \mathcal{P}$ ,  $P \neq P'$ .

Consequently, there is a *unique* shortest  $(s, t)$ -path  $P_{(s,t)}^*$  with respect to  $\lambda'$  for *all*  $(s, t) \in K_{\max}$ . We denote the set of these paths by

$$R := \{P_{(s,t)}^* : (s, t) \in K_{\max}\} .$$

Clearly,  $|R| = |K_{\max}|$  and  $S \subseteq R$ . □

Proposition 5.4 implies that the rank of the entire independence system  $(\mathcal{P}, \mathcal{I}_{USPS})$  is equal to  $|K_{\max}|$  and that all inclusion-wise maximal USPSs have the same cardinality, i.e.,

$$|S| = r(S) = r(\mathcal{I}_{USPS}) = |K_{\max}| \quad \text{for each maximal USPS } S.$$

In Section 5.5, we will show that computing the rank of an arbitrary path set  $S \subseteq \mathcal{P}$  is  $\mathcal{NP}$ -hard in general.

### 5.3 Properties of Unique Shortest Path Systems

One easily observes that each single path is a unique shortest path system. It is also obvious that, in a digraph  $D$  that contains only one path between each pair of nodes, any path set is a USPS. In general digraphs, however, not every path set is a USPS.

In this section, we discuss some fundamental properties of unique shortest path systems — the so-called Bellman property, which expresses the compatibility between pairs of paths, and some of its extensions.

#### 5.3.1 The Bellman Property

For a path  $P$  and two distinct nodes  $u, v \in P$ , we denote by  $P[u, v]$  the  $(u, v)$ -subpath of  $P$ . For simplicity, we also define  $P[u, v] = \emptyset$  if  $u$  or  $v$  do not occur in  $P$  at all or if  $u$  or  $v$  occur in  $P$  in reverse order.

Let  $P \in \mathcal{P}$  and consider a metric  $\lambda \in \mathbb{R}_+^A$  with respect to which  $P$  is the unique shortest  $(s_P, t_P)$ -path. Then clearly every  $(u, v)$ -subpath of  $P$  is the unique shortest  $(u, v)$ -path with respect to  $\lambda$  in  $D$ . The general principle that the optimality of some structure implies the optimality of certain substructures is often referred to as the *Bellman principle* or *Bellman property*. It was introduced by Bellman in the context of dynamic programming [21, 23, 24]. For unique shortest path systems, this principle implies a very fundamental property.

#### Definition 5.5

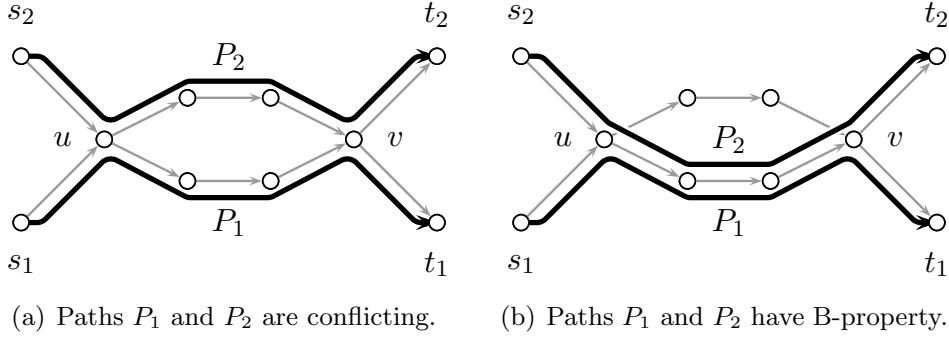
- (i) Two paths  $P_1, P_2 \in \mathcal{P}$  are said to have the **Bellman property (B-property)** if  $P_1[u, v] = P_2[u, v]$  holds for any pair of nodes  $u, v \in V$  with  $P_1[u, v] \neq \emptyset$  and  $P_2[u, v] \neq \emptyset$ . Otherwise, the paths  $P_1$  and  $P_2$  are called **conflicting**.
- (ii) A path set  $S \subseteq \mathcal{P}$  is said to have the **B-property** if it contains no pair of conflicting paths.

The Bellman property for pairs of paths is illustrated in Figure 5.3. In the literature, this property is also called *sub-path property* or *sub-optimality of paths*, c.f. [27, 180, 46].

**Proposition 5.6** *Any USPS has the B-property.*

**Proof.** Suppose that a USPS  $S \subseteq \mathcal{P}$  contains a pair of conflicting paths  $P_1$  and  $P_2$ . Then there are two nodes  $u$  and  $v$  such that  $P_1$  and  $P_2$  contain different  $(u, v)$ -subpaths  $P_1[u, v]$  and  $P_2[u, v]$ . Let  $\lambda$  be a compatible metric for  $S$ . As  $P_1$  is the unique shortest  $(s_{P_1}, t_{P_1})$ -path, we have

$$\begin{aligned} \lambda(P_1) &= \lambda(P_1[s_{P_1}, u]) + \lambda(P_1[u, v]) + \lambda(P_1[v, t_{P_1}]) \\ &< \lambda(P_1[s_{P_1}, u]) + \lambda(P_2[u, v]) + \lambda(P_1[v, t_{P_1}]) , \end{aligned}$$



**Figure 5.3** Bellman property for paths.

and consequently  $\lambda(P_1[u, v]) < \lambda(P_2[u, v])$ .

Analogously, the optimality of  $P_2$  implies  $\lambda(P_1[u, v]) > \lambda(P_2[u, v])$ . As these two inequalities contradict each other,  $S$  cannot contain a pair of conflicting paths.  $\square$

In general, the B-property is not sufficient to characterize all USPSs in an arbitrary digraph. It imposes a necessary condition for the existence of a compatible metric for a given path set  $S$ , but for path sets  $|S| \geq 4$  this condition is not sufficient. An example of a path set that has the B-property but that is no USPS is shown in Figure 5.4. USPSs with no more than three paths, however, are completely characterized by the B-property.

**Theorem 5.7** *A path set  $S \subseteq \mathcal{P}$  with  $|S| \leq 3$  is a USPS if and only if  $S$  has the B-property.*

**Proof.** In Proposition 5.6, we have already shown that any USPS has the B-property. Thus, it only remains to show that any path set  $S$  with  $|S| \leq 3$  that has the B-property is indeed a USPS.

For  $|S| = 1$  and  $|S| = 2$ , it is obvious that the metric

$$\lambda_a := \begin{cases} 1, & \text{if } a \in P \text{ for some } P \in S, \text{ and} \\ |V|, & \text{otherwise,} \end{cases}$$

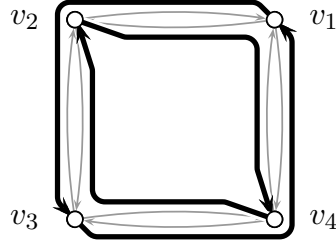
is compatible with  $S$ .

So, consider the case where  $S = \{P_1, P_2, P_3\}$  and let

$$\begin{aligned} A' &:= \{a \in A : a \in P_1 \cup P_2\}, \text{ and} \\ A'' &:= \{a \in A : a \in P_1 \cup P_2 \cup P_3\}. \end{aligned}$$

Since  $S$  has the B-property, so does  $\{P_1, P_2\}$ . Consequently,  $P_1$  is the only  $(s_{P_1}, t_{P_1})$ -path and  $P_2$  the only  $(s_{P_2}, t_{P_2})$ -path in the subgraph  $(V, A')$ .

Next, assume that the nodes in  $D$  are labeled  $v_1, \dots, v_{|V|}$  in such a way, that the nodes  $v_1$  to  $v_{|P_3|}$  appear in  $P_3$  in this order. In other words,  $P_3 = (v_1, \dots, v_{|P_3|})$ , as shown in Figure 5.5. Because  $S$  has the B-property, any subpath  $P_1[v_i, v_j]$  of  $P_1$  and any subpath  $P_2[v_i, v_j]$  of  $P_2$  with  $1 \leq i < j \leq |P_3|$



$$\begin{aligned} P_1 &= (v_1, v_2, v_3) \\ P_2 &= (v_3, v_4, v_1) \\ P_3 &= (v_2, v_1, v_4) \\ P_4 &= (v_4, v_3, v_2) \end{aligned}$$

**Figure 5.4** Non-USPS with B-property.

is a subpath of  $P_3$ , too. Hence,  $P_3$  is the only simple  $(v_1, v_{|P_3|})$ -path in the subgraph  $(V, A'')$ . With these two observations, it now is easy to verify that the metric  $\lambda \in \mathbb{Z}_+^A$  given by

$$\lambda_a := \begin{cases} 1, & a \in A', \\ |V|, & a \in A'', \text{ and} \\ |V|^2, & \text{otherwise} \end{cases}$$

is compatible with  $S$ . Consequently,  $S$  is a USPS.  $\square$

**Remark 5.8** If a path set  $S \subseteq \mathcal{P}$  has the B-property, then, for any  $v \in V$ , the paths  $P \in \mathcal{P}$  with source  $v$  form an arborescence rooted at  $v$  and, analogously, all paths with destination  $v$  form an anti-arborescence rooted at  $v$ . Path sets where all paths with the same destination form anti-arborescences (but which not necessarily have the B-property) correspond to so-called *unsplittable source-invariant routings*, a routing variant that is less restrictive than unsplittable shortest path routing and arises in data networks where the routers' forwarding tables are managed manually. In Chapter 6, we discuss this routing paradigm and its relation to unsplittable shortest path routing in more detail.

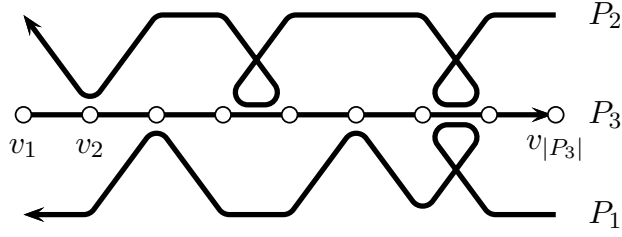
### 5.3.2 The Generalized Bellman Property

A stronger condition than the Bellman property can be derived by requiring the existence of a compatible path also for those node pairs for which no shortest path is prescribed and, furthermore, considering also the removal of arcs and nodes in the underlying digraph.

For any subgraph  $D'$  of  $D$ , we denote by  $\mathcal{P}_{D'}$  the set of all simple paths in  $D'$  and by  $\mathcal{P}_{D'}(s, t)$  the set of all simple  $(s, t)$ -paths in  $D'$ .

**Definition 5.9** Let  $S \subseteq \mathcal{P}$  be a path set in  $D$ .

- (i) The set  $S$  has the **generalized Bellman property (GB-property) with respect to a subgraph  $D'$  of  $D$**  if, for each pair of nodes  $s, t$  with  $\mathcal{P}_{D'}(s, t) \neq \emptyset$ , there exists a path  $P_{(s,t)}^* \in \mathcal{P}_{D'}(s, t)$  that does not conflict with any of the paths in  $S \cap \mathcal{P}_{D'}$ .



**Figure 5.5** Path set of size 3 with B-property.

- (ii) We say that  $S$  has the **GB-property** if it has the GB-property with respect to any subgraph  $D'$  of  $D$ .

The following example illustrates the generalized Bellman property.

**Example 5.2** Consider the digraph  $D = (V, A)$  with

$$\begin{aligned} V &:= \{v_1, \dots, v_6\} \quad \text{and} \\ A &:= \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_1, v_6), (v_6, v_5), \\ &\quad (v_5, v_4), (v_1, v_4), (v_2, v_5), (v_6, v_3)\} \end{aligned}$$

and the paths  $P_1 := (v_1, v_2, v_3)$ ,  $P_2 := (v_1, v_6, v_5)$ ,  $P_3 := (v_2, v_5, v_4)$ , and  $P_4 := (v_6, v_3, v_4)$ , as illustrated in Figure 5.6.

One easily finds that the path set  $S = \{P_1, \dots, P_4\}$  has the B-property. Even more, in the entire digraph  $D$ , there exists a  $(v_1, v_4)$ -path that does not conflict with the four paths  $P_1, \dots, P_4$ , namely  $P_5 = (v_1, v_4)$ .

In the subgraph  $D' := D - (v_1, v_4)$ , however, there exists no such path; any  $(v_1, v_4)$ -path in  $D'$  conflicts with at least one of the four paths  $P_1, \dots, P_4$ . Hence, the path set  $S$  does not have the generalized Bellman property with respect to the subgraph  $D'$ .  $\triangle$

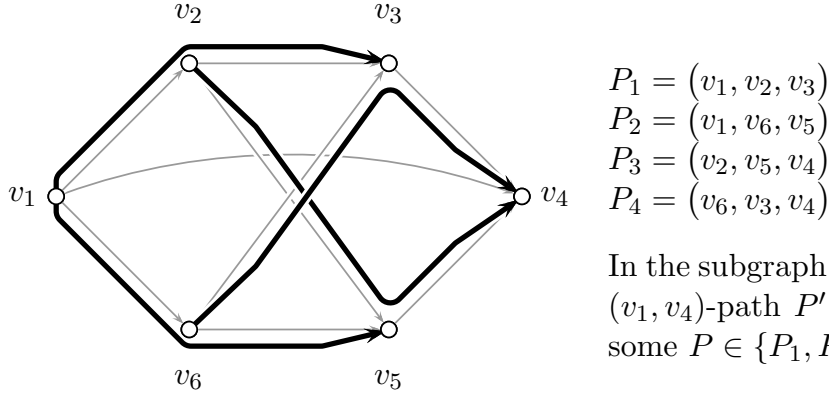
One easily observes that the GB-property implies the Bellman property, but — as shown in the above example — the Bellman property does not imply the GB-property. Any path set that has the GB-property (even if only with respect to the entire digraph  $D$ ) also has the B-property, but not vice versa. Removing the arc  $(v_1, v_4)$  from the digraph  $D$  in the above example yields a simple example where already the GB-property with respect to the entire digraph  $D$  is stronger than the Bellman property.

For undirected path sets, the GB-property is equivalent to the *generalized cyclic comparability* introduced by Ben-Ameur and Gourdin [27], see also Section 5.6.

**Proposition 5.10** *Any USPS has the GB-property.*

**Proof.** Let  $S \subseteq \mathcal{P}$  be a USPS in  $D$ , let  $D'$  be a subgraph of  $D$ , and let  $s$  and  $t$  be two nodes with  $\mathcal{P}_{D'}(s, t) \neq \emptyset$ . Applying Lemma 5.4 to the subgraph  $D'$ , we find that there exists a metric  $\lambda \in \mathbb{Z}_+^A$  that is compatible with  $S \cap \mathcal{P}_{D'}$





**Figure 5.6** Path set with B-property but without GB-property.

and, furthermore, induces a unique shortest path  $(s, t)$ -path  $P_{(s,t)}^*$  within the subgraph  $D'$  for all node pairs  $(s, t)$  with  $\mathcal{P}_{D'}(s, t) = \emptyset$ .

To prove Proposition 5.10, it is sufficient to show that none of the paths  $P \in S \cap \mathcal{P}_{D'}$  conflicts with  $P_{(s,t)}^*$ .

Suppose there is a path  $P \in S \cap \mathcal{P}_{D'}$  that conflicts with  $P_{(s,t)}^*$ . Then there are two nodes  $u$  and  $v$  such that  $P[u, v] \neq \emptyset$ ,  $P_{(s,t)}^*[u, v] \neq \emptyset$ , and  $P[u, v] \neq P_{(s,t)}^*[u, v]$ . Since  $P_{(s,t)}^*$  is the unique shortest  $(s, t)$ -path in  $D'$  with respect to  $\lambda$ , we have  $\lambda(P_{(s,t)}^*[u, v]) < \lambda(P[u, v])$ . On the hand, the fact that  $\lambda$  is compatible with  $S$  implies that  $\lambda(P[u, v]) < \lambda(P_{(s,t)}^*[u, v])$ . As these two inequalities contradict each other,  $P_{(s,t)}^*$  cannot conflict with any of the paths in  $P \in S \cap \mathcal{P}_{D'}$ .  $\square$

Also the GB-property is not sufficient to characterize the family of all USPSs, in general. For example, the path set shown in Figure 5.4 has the GB-property but is no USPS.

Both the B-property and the GB-property of unique shortest path systems correspond to structural properties of the independence system  $(\mathcal{P}, \mathcal{I}_{USPS})$  and the conflict hypergraph  $\mathcal{H}_{USPS}$ .

Each circuit of size two in  $(\mathcal{P}, \mathcal{I}_{USPS})$  and each simple edge (i.e., hyperedge of size two) in  $\mathcal{H}_{USPS}$  corresponds to a pair of conflicting paths, and vice versa. Theorem 5.7 implies that there are no circuits of size three in  $(\mathcal{P}, \mathcal{I}_{USPS})$  and, equivalently, no hyperedges of size three in  $\mathcal{H}_{USPS}$ .

Similarly, the GB-property characterizes a class of larger circuit and hyperedges in  $(\mathcal{P}, \mathcal{I}_{USPS})$  and  $\mathcal{H}_{USPS}$ . Any irreducible non-USPS with the B-property but without the GB-property is a circuit of size at least four in  $(\mathcal{P}, \mathcal{I}_{USPS})$  and corresponds to a non-simple hyperedges in  $\mathcal{H}_{USPS}$ . The path set shown in Figure 5.4, for instance, is such an irreducible non-USPS.

### 5.3.3 Further Properties

Further properties of unique shortest path systems can be derived by analyzing the infeasibilities that may potentially arise in the linear programming formulations of the INVERSE UNIQUE SHORTEST PATHS problem.

According to Theorem 4.4 in Chapter 4, a path set  $S \subseteq \mathcal{P}$  is a non-USPS if and only if the corresponding linear inequality system

$$\sum_{a \in P'} \lambda_a - \sum_{a \in P} \lambda_a \geq 1 \quad \forall P \in S, P' \in \mathcal{P}(s_P, t_P) \setminus \{P\}, \quad (4.3)$$

$$\lambda_a \geq 1 \quad \forall a \in A \quad (4.4)$$

is infeasible. Applying Farkas' lemma [87, 88, 175] to this linear system, one finds that  $S$  is a non-USPS if and only if there exist dual multipliers  $\mu_{P,Q} \geq 0$  for all  $P \in S$  and  $Q \in \mathcal{P}(s_P, t_P) \setminus \{P\}$  that satisfy

$$\begin{aligned} \sum_{\substack{P \in S, Q \in \mathcal{P}(s_P, t_P): \\ Q \neq P, a \in P}} \mu_{P,Q} - \sum_{\substack{P \in S, Q \in \mathcal{P}(s_P, t_P): \\ Q \neq P, a \in Q}} \mu_{P,Q} \leq 0 \quad \forall a \in A, \text{ and} \\ \sum_{\substack{P \in S, Q \in \mathcal{P}(s_P, t_P): \\ Q \neq P}} \mu_{P,Q} > 0. \end{aligned}$$

A vector  $\mu$  of such multipliers is called a **(dual) Farkas ray** for (4.3)–(4.4).

In principle, the combinatorial properties of unique shortest path systems can be deduced from the Farkas rays for (4.3)–(4.4) (or, equivalently, the Farkas rays for the linear system (4.11)–(4.14) discussed in Section 5.7). Since any path set  $S$  that is not a USPS leads to an infeasible system (4.3)–(4.4), all irreducible non-USPSs can be derived from the possible extremal Farkas rays.

Alternatively, the potentially arising irreducibly inconsistent subsystems of (4.3)–(4.4) (or (4.11)–(4.14)) may be analyzed. An **irreducible inconsistent subsystem (IIS)** of a linear program is a subset of the inequalities, such that this set of inequalities has no solution but every proper subset of it has. An inclusion-wise minimal such infeasible set of inequalities can be computed efficiently using a combination of Greedy and linear programming techniques, but computing a minimum cardinality such set however is  $\mathcal{NP}$ -hard. A comprehensive survey on IISs and related problems can be found in [161], for example.

One easily observes that each irreducible path set that violates the B-property corresponds to an IISs of size two or, equivalently, to a Farkas ray with only two non-zero entries in (4.3)–(4.4), and vice versa. Similarly, any irreducible non-USPS that satisfies the B-property but violates the GB-property corresponds to an IIS consisting of more than two inequalities and to a Farkas ray with more than two non-zero entries.

Broström and Holmberg [46] derived another property of USPS by analyzing a special class of infeasibilities that may occur in (a variation of)

the linear system (4.11)–(4.14). This so-called *valid-cycle property* can be expressed as the non-existence of an augmenting cycle in a related multicommodity flow problem. It is stronger than the B-property and can be verified in polynomial time for any given path set. However, also this property is not sufficient to completely characterize the family of all USPSs in general digraph. (Originally, Broström and Holmberg derived the valid-cycle property for path sets that correspond to multi-shortest path routings, but their results carry over to unique shortest path systems straightforward.)

**Remark 5.11** It follows from Lemma 4.5 in Chapter 4 and the polynomial time equivalence of optimization and separation [103], that the linear system (4.3)–(4.4) can be either solved or proven to be infeasible in polynomial time. Thus, for any non-USPS  $S$ , there exist a Farkas ray with only polynomially many non-zero entries and an IIS with only polynomially many inequalities.

## 5.4 Finding Irreducible Non-Unique Shortest Path Systems

A dual Farkas ray or an irreducible inconsistent system of (4.3)–(4.4) are adequate proofs that a given path set  $S \subseteq \mathcal{P}$  is no USPS. In practical applications, however, these certificates are not very helpful. Typically, we are more interested in finding an irreducible non-USPS that is contained in the given non-USPS. Such a set describes an (inclusion-wise) minimal conflict among the given paths and thereby tells which paths have to be modified to turn the path set into a USPS.

Often, we not only seek for an inclusion-wise minimal such non-USPS, but we wish to find a non-USPS of minimum cardinality or minimum weight. The first task can be formulated as the following optimization problem:

<i>Problem:</i>	MIN-NON-USPS
<i>Instance:</i>	A digraph $D = (V, A)$ and a non-USPS $S \subseteq \mathcal{P}$ .
<i>Solution:</i>	An irreducible non-USPS $R \subseteq S$ .
<i>Objective:</i>	$\min  R $ .

The more general task of finding an irreducible non-USPS of minimum weight within a given set of paths can be formalized as follows:

<i>Problem:</i>	MIN-WEIGHT-NON-USPS
<i>Instance:</i>	A digraph $D = (V, A)$ , a non-USPS $S \subseteq \mathcal{P}$ , and strictly positive weights $w \in \mathbb{Z}_+^S$ .
<i>Solution:</i>	An irreducible non-USPS $R \subseteq S$ .
<i>Objective:</i>	$\min \sum_{P \in S} w_P$ .

Clearly, both problems MIN-NON-USPS and MIN-WEIGHT-NON-USPS belong to the class of  $\mathcal{NP}$  optimization problems  $\mathcal{NPO}$ .

Note that in both problems we seek for an irreducible non-USPS that is contained within a given path set  $S$ . The weight function  $\lambda$  in MIN-WEIGHT-NON-USPS is defined only for those paths that are contained in  $S$  (which is equivalent to saying that  $\lambda$  is given as a sparse-vector that defines a weights value only for the paths in  $S$ , and all other weights are assumed to be infinity). For both problems the encoding size of an instance and the encoding size of a solution thus depend on the size of the given path set  $S$ , but neither the size of an instance nor the size of a solution is necessarily exponential in the size of the underlying digraph.

We also assume that the given path set  $S$  is a non-USPS and that all weights are strictly positive in the weighted case. These restrictions are necessary only to ensure that the optimal solution value is always strictly positive, which is technically important when discussing approximation algorithms.

The weighted problem version MIN-WEIGHT-NON-USPS will be of special interest later in this thesis. It is closely related to the separation problem for (irreducible) non-USPS inequalities for an integer linear programming formulation of unsplittable shortest path routing problems, which is discussed in Chapter 7.

### 5.4.1 Greedy Approaches

The most natural approach to solve MIN-NON-USPS and MIN-WEIGHT-NON-USPS is to use the greedy algorithm for finding circuits in independence systems. Starting with the given non-USPS  $S$ , this algorithm iteratively tries to remove each path from the current path set. If the remaining path set still is a non-USPS, the current path is removed permanently and the algorithm continues with the next path and the reduced path set. If removing a path from the current path set would yield a USPS, then this path remains in the set and the algorithm continues with the next path to be removed. Whether the removal of a path yields a USPS or a non-USPS can be determined by solving the corresponding INVERSE UNIQUE SHORTEST PATHS (IUSP) problem. The algorithm finishes after all paths have been tried for removal and returns the remaining path set.

Clearly, the path set finally returned by the greedy algorithm is an irreducible non-USPS. It is also easy to see that this algorithm can be implemented to run in polynomial time: It performs only  $|S|$  iterations, and each iteration's IUSP problem can be solved in polynomial time with respect size of the underlying digraph and the candidate path set using the linear programming techniques discussed in Chapter 4.

The standard version of this greedy algorithm removes only one path in each iteration. Independent of the size of the irreducible non-USPS returned

in the end, the algorithm performs  $|S|$  many iterations, each involving the solution of a linear inequality system. This is not very efficient in practice, because the size of an irreducible non-USPS is typically very small compared to the size of the given path set. A substantial speed-up can be achieved by exploiting the information encoded in the dual Farkas rays computed as a by-product when solving the (infeasible) linear system.

**Observation 5.12** *Let  $S \subseteq \mathcal{P}$  be a non-USPS and let  $\mu$  be a dual Farkas ray for the corresponding linear system (4.3)–(4.4). Then the path set*

$$\text{con}(\mu) := \{P \in S : \mu_{P,Q} > 0 \text{ for some } Q \in \mathcal{P}(s_P, t_P) \setminus \{P\}\}$$

*is a non-USPS.*

Algorithm Greedy-Non-USPS shown below demonstrates how Observation 5.12 can be used to improve the standard greedy approach: Whenever the linear system (4.3)–(4.4) is infeasible for the current path set, we replace the current path set by the non-USPS  $R := \text{con}(\mu)$  for the corresponding dual Farkas ray  $\mu$ .

**Theorem 5.13** *Algorithm Greedy-Non-USPS computes an irreducible non-USPS  $R \subseteq S$  or proves that  $S$  is a USPS in polynomial time.*

**Proof.** Follows immediately from Observation 5.12. □

In practice, the Farkas ray based substitution proved to be very effective. It often leads to a drastic reduction of the candidate path set in the first few iterations of the algorithm and, thereby, to much fewer iterations in total. However, the worst case number of iterations of algorithm Greedy-Non-USPS remains  $|S|$ .

---

**Algorithm 5.1** Greedy-Non-USPS

---

**Input:** Path set  $S \subseteq \mathcal{P}$ .

**Output:** An irreducible non-USPS  $R \subseteq S$  or assertion that  $S$  is a USPS.

1. If (4.3)–(4.4) is *feasible* for path set  $S$  then  
Return ' $S$  is a USPS'.
  2. Let  $\mu$  be a dual Farkas ray of (4.3)–(4.4) for  $S$ .  
Set  $R := \text{con}(\mu)$ .
  3. For each  $P \in R$  do (in order of non-increasing weights  $w_P$ )
    - 3.1 If (4.3)–(4.4) is *infeasible* for  $R \setminus \{P\}$  then
      - 3.2 Let  $\mu$  be a corresponding dual Farkas ray.  
Set  $R := \text{con}(\mu)$ .
  4. Return  $R$ .
-

Analogously, one can exploit the information encoded in an irreducible inconsistent subsystem (IIS) of the infeasible linear system (4.3)–(4.4) to reduce the size of the candidate path set in Step 2 and in Step 3.2 of algorithm Greedy-Non-USPS using the following observation.

**Observation 5.14** *Let  $S \subseteq \mathcal{P}$  be a non-USPS and let  $I$  be an (irreducible) inconsistent subsystem of inequalities of the corresponding linear system (4.3)–(4.4). Then the path set*

$$\text{con}(I) := \{ P \in S : I \text{ contains inequality (4.3) for } P \text{ and} \\ \text{some } Q \in \mathcal{P}(s_P, t_P) \setminus \{P\} \}$$

*is a non-USPS.*

Note that the path set  $\text{con}(I)$  defined in Observation 5.14 is only a “normal” non-USPS in general. Even if  $I$  is an *irreducible* inconsistent subsystem of rows, the corresponding path set  $\text{con}(I)$  is *not* necessarily an *irreducible* non-USPS.

Algorithm Greedy-Non-USPS can be turned into an approximation algorithm for the weighted problem version MIN-WEIGHT-NON-USPS by considering the paths in Step 3 in order of non-increasing costs  $w_P$ . One easily observes that this algorithm achieves a worst-case performance ratio of  $|S|$  for both problem versions MIN-NON-USPS and MIN-WEIGHT-NON-USPS. It is also not difficult to construct classes of examples where a ratio of  $\Theta(|S|)$  is attained.

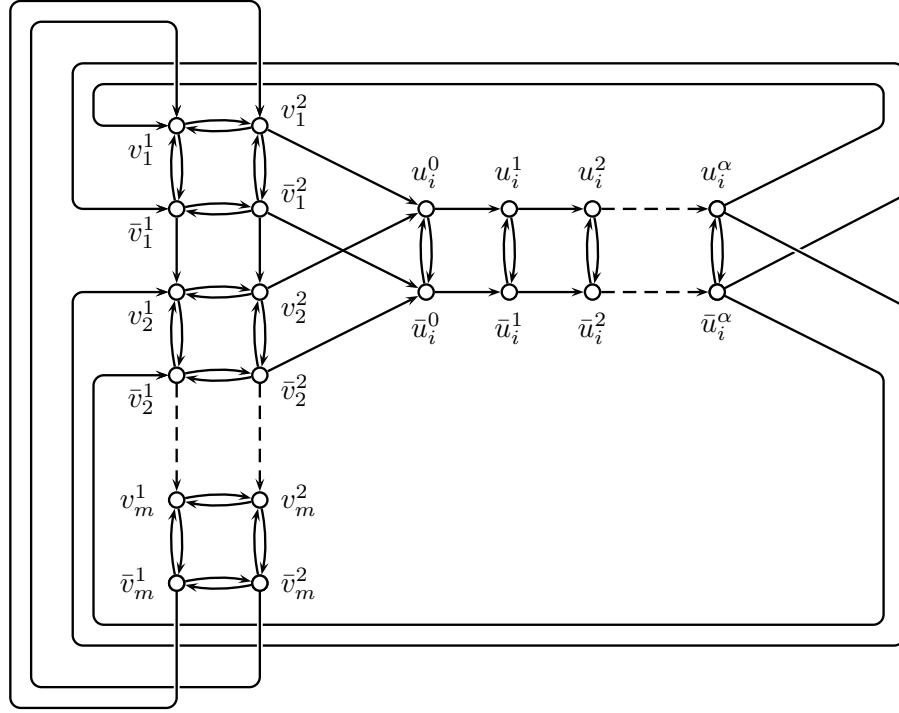
### 5.4.2 Inapproximability Results

For any fixed  $k \in \mathbb{Z}_+$ , we can easily decide in polynomial time whether there exists an irreducible non-USPS  $R \subseteq S$  with  $|R| \leq k$  or find a minimum weight such non-USPS by checking all subsets  $R \subseteq S$  with  $|R| \leq k$ . In the special case where the size of the possible irreducible non-USPSes is bounded by some constant, MIN-NON-USPS and MIN-WEIGHT-NON-USPS thus are solvable in polynomial time.

In general, however, it is already computational hard to approximate these problems within a factor less than  $7/6$ .

**Theorem 5.15** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate MIN-NON-USPS within a factor of  $7/6 - \epsilon$ . This holds even if each path  $P \in S$  is a shortest  $(s_P, t_P)$ -path (w.r.t. the number of arcs) with  $|P| = 2$ .*

**Proof.** We construct an approximation preserving reduction from the optimization problem MINIMUM VERTEX COVER to MIN-NON-USPS. The problem MINIMUM VERTEX COVER is defined as follows: Given an undirected graph  $H = (W, E)$ , find a minimum cardinality set  $C \subseteq W$  such that, for each edge  $uv \in E$ , at least one of the nodes  $u$  and  $v$  belongs to  $C$ . This



**Figure 5.7** Subgraph of  $D$  corresponding to the edges  $f_1, \dots, f_m$ , node  $w_i \in W$ , and incidences  $f_1, f_2 \in \delta(w_i)$ .

problem was shown to be  $\mathcal{APX}$ -complete by Papadimitriou and Yannakakis [155, 156] and to be inapproximable within  $7/6 - \epsilon$  for any  $\epsilon > 0$  by Håstad [112]. Its associated decision problem VERTEX COVER is one of classical  $\mathcal{NP}$ -complete problems introduced by Karp [121] and Garey and Johnson [96].

Suppose we are given an instance  $H = (W, F)$  of MINIMUM VERTEX COVER consisting of the nodes  $w_i$  with  $i \in I := \{1, \dots, n\}$  and the edges  $f_k$  with  $k \in K = \{1, \dots, m\}$ . Let  $\alpha \in \mathbb{Z}_+$  be a large integer number. At the end of the proof, we discuss how to choose  $\alpha$  appropriately. We construct a MIN-NON-USPS instance consisting of a digraph  $D = (V, A)$  and a path set  $S \subseteq \mathcal{P}_D$  as follows.

For each  $i \in I$ , we introduce  $2\alpha + 2$  nodes  $u_i^0, \dots, u_i^\alpha$  and  $\bar{u}_i^0, \dots, \bar{u}_i^\alpha$ . These nodes are connected by the arcs  $(u_i^l, \bar{u}_i^l)$  and  $(\bar{u}_i^l, u_i^l)$  with  $l = 0, \dots, \alpha$ , and by the arcs  $(u_i^l, u_i^{l+1})$  and  $(\bar{u}_i^l, \bar{u}_i^{l+1})$  with  $l = 0, \dots, \alpha - 1$ .

For each  $k \in K$ , we add four nodes  $v_k^1, v_k^2, \bar{v}_k^1$ , and  $\bar{v}_k^2$ . These are interconnected by the arcs  $(v_k^1, v_k^2)$ ,  $(v_k^2, v_k^1)$ ,  $(\bar{v}_k^1, \bar{v}_k^2)$ ,  $(\bar{v}_k^2, \bar{v}_k^1)$ ,  $(v_k^1, \bar{v}_k^1)$ ,  $(\bar{v}_k^1, v_k^1)$ ,  $(v_k^2, \bar{v}_k^2)$ , and  $(\bar{v}_k^2, v_k^2)$  for all  $k \in K$ . Furthermore, we add two arcs  $(\bar{v}_k^1, v_{k+1}^1)$  and  $(\bar{v}_k^2, v_{k+1}^2)$  for each  $k = 1, \dots, m - 1$ , and two arcs  $(\bar{v}_m^1, v_1^1)$  and  $(\bar{v}_m^2, v_1^2)$  for  $k = m$ .

Finally, we introduce arcs for the node-edge incidences in  $H$ . For all  $i \in I$  and all  $k$  with  $f_k \in \delta(w_i)$ , we add the four arcs  $(v_k^2, u_i^0)$ ,  $(\bar{v}_k^2, \bar{u}_i^0)$ ,  $(u_i^\alpha, v_k^1)$ ,

and  $(\bar{u}_i^\alpha, \bar{v}_k^1)$ . The resulting digraph  $D$  is illustrated in Figure 5.7.

The path set  $S$  consists of four different types of paths. For each  $i \in I$ , it contains the  $2\alpha$  many paths

$$S_i^1 := \{(\bar{u}_i^l, u_i^l, u_i^{l+1}), (u_i^l, \bar{u}_i^l, \bar{u}_i^{l+1}) \mid l = 0, \dots, \alpha - 1\}.$$

For each  $k \in K$ ,  $S$  contains four paths. For  $k \neq m$ , it contains the paths

$$S_k^2 := \{(\bar{v}_k^1, v_k^1, v_k^2), (v_k^2, v_k^1, \bar{v}_k^1), (\bar{v}_k^1, \bar{v}_k^2, v_{k+1}^2), (\bar{v}_k^2, \bar{v}_k^1, v_{k+1}^1)\},$$

while for  $k = m$  it contains

$$S_m^2 := \{(\bar{v}_k^1, v_k^1, v_k^2), (v_k^2, v_k^1, \bar{v}_k^1), (\bar{v}_k^1, \bar{v}_k^2, v_{k+1}^1), (\bar{v}_k^2, \bar{v}_k^1, v_{k+1}^2)\}.$$

Finally,  $S$  contains four more paths for each node-edge incidence in  $H$ . For each  $i \in I$  and each  $k$  with  $f_k \in \delta(w_i)$ , these paths are

$$\begin{aligned} S_{i,k}^3 &:= \{(v_k^2, \bar{v}_k^2, \bar{u}_i^0), (\bar{v}_k^2, v_k^2, u_i^0)\} \text{ and} \\ S_{i,k}^4 &:= \{(u_i^\alpha, \bar{u}_i^\alpha, \bar{v}_k^1), (\bar{u}_i^\alpha, u_i^\alpha, v_k^1)\}. \end{aligned}$$

The path set  $S$  is the union of these sets, i.e.,

$$S := \bigcup_i S_i^1 \cup \bigcup_k S_k^2 \cup \bigcup_{i,k: f_k \in \delta(w_i)} S_{i,k}^3 \cup S_{i,k}^4.$$

Figure 5.8 illustrates these paths. Note that each path in  $S$  contains exactly two arcs and is a shortest path between its terminal nodes w.r.t. the number of arcs.

In the first part of the proof, we show that any vertex cover  $C \subseteq W$  in  $H$  can be transformed into a non-USPS  $R := R(C) \subseteq S$  in  $D$  with  $|R| = 2\alpha|C| + 8m$ . As a byproduct, this also proves that the constructed path system  $S$  is indeed a non-USPS.

Let  $C \subseteq W$  be a vertex cover in  $H$ . For each edge  $f = w_i w_j \in F$ , we denote  $c(f) := w_i$ , if  $i < j$  and  $w_i \in C$ , and  $c(f) := w_j$ , otherwise. Since  $C$  is a vertex cover,  $c(f) \in C$  for all  $f \in F$ . We define the path set corresponding to  $C$  as

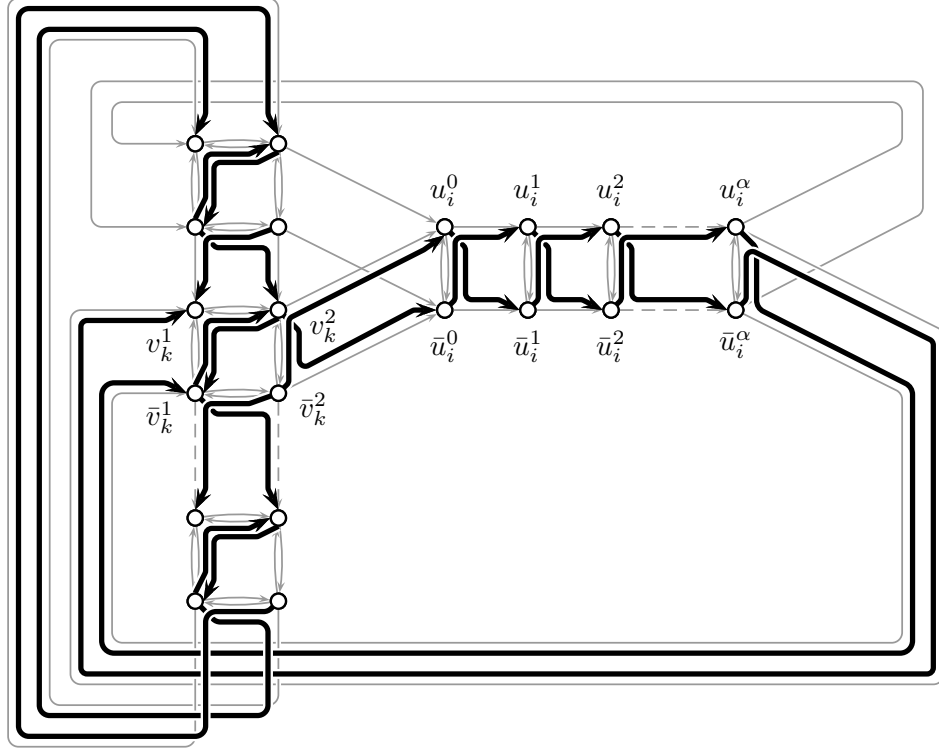
$$R = R(C) := \bigcup_k S_k^1 \cup \bigcup_{i: w_i \in C} S_i^2 \cup \bigcup_{i,k: w_i = c(f_k)} S_{i,k}^3 \cup S_{i,k}^4.$$

The size of this path set is

$$|R| = 2\alpha \cdot |C| + 8m. \quad (5.2)$$

It follows from Proposition 4.2 in Chapter 4 that  $R$  is a USPS if and only if the linear system (4.1)–(4.2) has a feasible solution for  $R$ . In order to show





**Figure 5.8** Union of the path sets  $S_k^2$  for all  $k \in K$  and the path sets  $S_i^1$ ,  $S_{i,k}^3$ , and  $S_{i,k}^4$  for some  $i, k$  with  $f_k \in \delta(w_i)$ .

that  $R$  is a non-USPS, it therefore is sufficient to show that the following sub-system of (4.1)–(4.2) has no solution:

$$\lambda(\bar{v}_k^1, v_k^1) + \lambda(v_k^1, v_k^2) < \lambda(\bar{v}_k^1, \bar{v}_k^2) + \lambda(\bar{v}_k^2, v_k^2) \quad \forall k \quad (5.3)$$

$$\lambda(v_k^1, \bar{v}_k^1) + \lambda(v_k^2, v_k^1) < \lambda(\bar{v}_k^2, \bar{v}_k^1) + \lambda(v_k^2, \bar{v}_k^2) \quad \forall k \quad (5.4)$$

$$\lambda(\bar{v}_k^1, \bar{v}_k^2) + \lambda(\bar{v}_k^2, v_{k+1}^2) < \lambda(\bar{v}_k^1, v_{k+1}^1) + \lambda(v_{k+1}^1, v_{k+1}^2) \quad \forall k \neq m \quad (5.5)$$

$$\lambda(\bar{v}_k^2, \bar{v}_k^1) + \lambda(\bar{v}_k^1, v_{k+1}^1) < \lambda(\bar{v}_k^2, v_{k+1}^2) + \lambda(v_{k+1}^2, v_{k+1}^1) \quad \forall k \neq m \quad (5.6)$$

$$\lambda(\bar{v}_m^1, \bar{v}_m^2) + \lambda(\bar{v}_m^2, v_1^1) < \lambda(\bar{v}_m^1, v_1^2) + \lambda(v_1^2, v_1^1) \quad (5.7)$$

$$\lambda(\bar{v}_m^2, \bar{v}_m^1) + \lambda(\bar{v}_m^1, v_1^2) < \lambda(\bar{v}_m^2, v_1^1) + \lambda(v_1^1, v_1^2) \quad (5.8)$$

$$\lambda(\bar{u}_i^l, u_i^l) + \lambda(u_i^l, u_i^{l+1}) < \lambda(\bar{u}_i^l, \bar{u}_i^{l+1}) + \lambda(\bar{u}_i^{l+1}, u_i^{l+1}) \quad \forall i, l : w_i \in C, l \neq \alpha \quad (5.9)$$

$$\lambda(u_i^l, \bar{u}_i^l) + \lambda(\bar{u}_i^l, \bar{u}_i^{l+1}) < \lambda(u_i^l, u_i^{l+1}) + \lambda(u_i^{l+1}, \bar{u}_i^{l+1}) \quad \forall i, l : w_i \in C, l \neq \alpha \quad (5.10)$$

$$\lambda(\bar{v}_k^2, v_k^2) + \lambda(v_k^2, u_i^0) < \lambda(\bar{v}_k^2, \bar{u}_i^0) + \lambda(\bar{u}_i^0, u_i^0) \quad \forall k, i : w_i = c(f_k) \quad (5.11)$$

$$\lambda(v_k^2, \bar{v}_k^2) + \lambda(\bar{v}_k^2, \bar{u}_i^0) < \lambda(v_k^2, u_i^0) + \lambda(u_i^0, \bar{u}_i^0) \quad \forall k, i : w_i = c(f_k) \quad (5.12)$$

$$\lambda(u_i^\alpha, \bar{u}_i^\alpha) + \lambda(\bar{u}_i^\alpha, \bar{v}_k^1) < \lambda(u_i^\alpha, v_k^1) + \lambda(v_k^1, \bar{v}_k^1) \quad \forall k, i : w_i = c(f_k) \quad (5.13)$$

$$\lambda(\bar{u}_i^\alpha, u_i^\alpha) + \lambda(u_i^\alpha, v_k^1) < \lambda(\bar{u}_i^\alpha, \bar{v}_k^1) + \lambda(\bar{v}_k^1, v_k^1) \quad \forall k, i : w_i = c(f_k) \quad (5.14)$$

Inequalities (5.3) ensure that, for each  $k$ , the path  $(\bar{v}_k^1, v_k^1, v_k^2)$  is strictly shorter than the other two-arc path  $(\bar{v}_k^1, \bar{v}_k^2, v_k^2)$  from  $\bar{v}_k^1$  to  $v_k^2$ . Together, (5.3)–(5.8) express that each path in  $\bigcup_k S_k^2$  is strictly shorter than its alternative other two-arc path. Analogously, inequalities (5.9) and (5.10) enforce that each path in  $\bigcup_{i:w_i \in C} S_i^1$  is shorter than its respective alternative two-

arc path, and inequalities (5.11)–(5.14) ensure this property for all paths in  $\bigcup_{k,i: w_i=c(f_k)} S_{i,k}^3 \cup S_{i,k}^4$ .

To verify that this system of strict linear inequalities has no solution, we apply Farkas' Lemma. For each  $i$  with  $w_i \in C$ , let  $\mu(i) := |\{f \in F : w_i = c(f)\}|$ . The dual multipliers  $\mu(i)$  for all inequalities (5.9) and (5.10) and 1 for all other inequalities certify that (5.3)–(5.14) is infeasible: Multiplying each inequality (5.9) and (5.10) with  $\mu(i)$  and then adding up these and all other inequalities of (5.3)–(5.14) yield a strict inequality that contains each variable  $\lambda_a$ ,  $a \in A$ , with the same coefficient on the left and on the right hand side. Consequently,  $R$  is not a USPS.

In the second part of the proof, we show that any irreducible non-USPS  $R \subseteq S$  in  $D$  can be transformed back into a vertex cover  $C := C(R) \subseteq W$  in  $H$  with  $2\alpha|C| \geq |R| - 8m$ . It is sufficient to define such a backward transformation only for irreducible non-USPSs, because any non-USPS  $R' \subseteq S$  in  $D$  can be reduced to an irreducible non-USPS  $R \subseteq R'$  in polynomial time using the greedy algorithm presented at the beginning of this section.

In order to define the backward transformation properly, we first need to show that all irreducible non-USPSs in  $D$  have a structure that is similar to that of the non-USPSs  $R(C)$  constructed in the first part of the proof. So, let  $R \subseteq S$  be an irreducible non-USPS.

First, observe that all paths in  $\bigcup_k S_k^2$  must be contained in  $R$ . Suppose there is some  $k'$  such that the path  $(v_{k'}^2, v_{k'}^1, \bar{v}_{k'}^1)$  does not belong to  $R$ . W.l.o.g., we may assume that  $k' = 1$ . Let  $M \geq 2|A|$  and consider the metric

$$\lambda_a := \begin{cases} M + l + 1, & \text{if } a = (u_i^l, \bar{u}_i^l) \text{ or } a = (\bar{u}_i^l, u_i^l), \\ M + \alpha + 2, & \text{if } a = (v_k^1, \bar{v}_k^1) \text{ or } a = (\bar{v}_k^1, v_k^1), \\ M + 1, & \text{if } a = (\bar{v}_k^1, v_{k+1}^1) \text{ or } a = (\bar{v}_k^2, v_{k+1}^2) \\ M + 1, & \text{if } a = (\bar{v}_m^1, v_1^2) \text{ or } a = (\bar{v}_m^2, \bar{v}_1^1) \\ M + (2m - 1)(\alpha + 4) + 1, & \text{if } a = (v_1^2, v_1^1), \\ M + (k - 2)(\alpha + 4) + 1, & \text{if } a = (v_k^2, v_k^1) \text{ for } k \neq 1, \\ M + (k - 1)(\alpha + 4), & \text{if } a = (\bar{v}_k^2, \bar{v}_k^1), \\ M + (m + k - 2)(\alpha + 4) + 1, & \text{if } a = (v_k^1, v_k^2), \\ M + (m + k - 1)(\alpha + 4), & \text{if } a = (\bar{v}_k^1, \bar{v}_k^2), \text{ and} \\ M, & \text{otherwise.} \end{cases}$$

One easily finds that  $M \leq \lambda_a < 3/2M$  for all  $a \in A$ . Since all paths in  $S$  contain two arcs, no path with three or more arcs in  $D$  can be shorter than any path in  $S$ . For each path  $P \in S \setminus \{(v_1^2, v_1^1, \bar{v}_1^1)\}$ , however, there is only one alternative  $(s_P, t_P)$ -path with only two arcs, and it is straightforward to verify that each path in  $S \setminus \{(v_1^2, v_1^1, \bar{v}_1^1)\}$  is indeed shorter than the corresponding alternative path. Hence,  $S \setminus \{(v_1^2, v_1^1, \bar{v}_1^1)\}$  is a USPS, which implies

that the path  $(v_1^2, v_1^1, \bar{v}_1^1)$  must be contained in the (irreducible) non-USPS  $R \subseteq S$ . Analogously, it follows that any other path  $P \in \bigcup_k S_k^2$  is contained in  $R$ .

With the same technique, we can show that, for any  $k \in K$ , there exist some  $i := c(k)$  with  $f_k \in \delta(w_i)$  such that  $S_{i,k}^3 \subset R$ : W.l.o.g. let  $k = 1$  and  $f_1 = w_i w_j$ , and suppose that none of the two paths  $(v_1^2, \bar{v}_1^2, \bar{u}_i^0)$  and  $(\bar{v}_1^2, v_1^2, u_j^0)$  belongs to  $R$ . Then, with  $\lambda$  defined as above, the metric

$$\lambda'_a := \begin{cases} \lambda_a + 2m(\alpha + 4) & , \text{ if } a = (v_1^2, \bar{v}_1^2) \text{ or } a = (v_1^2, u_j^0), \\ \lambda_a & , \text{ otherwise,} \end{cases}$$

is compatible with  $S \setminus \{(v_1^2, \bar{v}_1^2, \bar{u}_i^0), (\bar{v}_1^2, v_1^2, u_j^0)\}$ . Hence, the non-USPS  $R$  must contain both paths in  $S_{i,1}^3$  or both paths in  $S_{j,1}^3$ .

Analogously, one can show that  $S_i^1 \subset R$  for each  $k \in K$  and  $i = c(k)$  and, furthermore, that for any  $k \in K$  there exist some  $j = c'(k)$  with  $f_k \in \delta(w_j)$  such that  $S_{j,k}^4 \subset R$ .<sup>1)</sup>

Now we can define the vertex set corresponding to the irreducible non-USPS  $R$  as

$$C = C(R) := \{w_{c(k)} : k \in K\}.$$

Because  $c(k)$  is either  $i$  or  $j$  for any edge  $f_k = w_i w_j \in F$ , the set  $C$  is a vertex cover in  $H$ . The above observations imply that

$$\begin{aligned} |R| &\geq \sum_{k \in K} \left( |S_k^2| + |S_{c(k),k}^3| + |S_{c'(k),k}^4| \right) + \sum_{i \in I: w_i \in C} |S_i^W| \\ &\geq 8m + 2\alpha |C|. \end{aligned} \tag{5.15}$$

Now, it follows straightforward that MIN-NON-USPS is at least as hard to approximate as MINIMUM VERTEX COVER. Suppose there is an  $(r - \epsilon)$ -approximation algorithm **ALG** for MIN-NON-USPS with  $\epsilon > 0$  and  $(r - \epsilon) \geq 1$ . Then we choose  $\alpha := \lceil 8m(r - 1 - \epsilon)/\epsilon \rceil$ . With this choice of  $\alpha$ , both the construction of the MIN-NON-USPS instance as well as the backward transformation of an irreducible non-USPS to a vertex cover are polynomial in the encoding size of  $H$ .

Due to (5.2) and (5.15), we have  $|R^*| = 8m + 2\alpha |C^*|$  for any minimum vertex cover  $C^*$  in  $H$  and any minimum non-USPS  $R^* \subseteq S$  in  $D$ . Furthermore,

$$\frac{2\alpha |C(R)| + 8m}{2\alpha |C^*| + 8m} \leq \frac{|R|}{|R^*|} \leq r - \epsilon$$

---

<sup>1)</sup>Note that  $i = c(k)$  and  $j = c'(k)$  may be different. There may exist an irreducible non-USPS  $R \subseteq S$  in the constructed digraph  $D$  that contains only one of the two path sets  $S_{i,k}^3$  and  $S_{i,k}^4$  for each  $f_k \in \delta(w_i)$ .

implies that

$$2\alpha|C(R)| \leq 2\alpha|C^*|(r - \epsilon) + 8m(r - 1 - \epsilon) ,$$

and, further on,

$$\frac{|C(R)|}{|C^*|} \leq r - \epsilon + \frac{8m(r - 1 - \epsilon)}{2\alpha|C^*|} \leq r - \epsilon + \frac{\epsilon}{2|C^*|} \leq r - \frac{\epsilon}{2} .$$

Put in words, any  $(r - \epsilon)$ -approximate solution  $R \subseteq S$  for the constructed MIN-NON-USPS instance can be transformed back to a  $(r - \epsilon/2)$ -approximate solution  $C(R)$  of the given MINIMUM VERTEX COVER instance. Thus, any  $(r - \epsilon)$ -approximation algorithm for MIN-NON-USPS yields an  $(r - \epsilon/2)$ -approximation algorithm for MINIMUM VERTEX COVER. Since MINIMUM VERTEX COVER is  $\mathcal{NP}$ -hard to approximate within a factor strictly less than  $7/6$ , so is MIN-NON-USPS.  $\square$

The constant inapproximability threshold for MIN-NON-USPS carries over directly to the weighted problem MIN-WEIGHT-NON-USPS.

**Corollary 5.16** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate MIN-WEIGHT-NON-USPS within a factor of  $7/6 - \epsilon$ .*

One easily verifies that Corollary 5.16 also holds if the path weights are given implicitly by an arc weight vector  $c \in \mathbb{Z}_+^A$  as  $w_P := \sum_{a \in P} c_a$  for each path  $P \in \mathcal{P}$ .

In terms of the independence system  $(\mathcal{P}, \mathcal{I}_{USPS})$  and conflict hypergraph  $\mathcal{H}_{USPS}$ , Corollary 5.16 states that the problem of finding a minimum weight circuit in  $(\mathcal{P}, \mathcal{I}_{USPS})$  that is contained in a given (dependent) set and, equivalently, the problem of finding a minimum weight hyperedge in  $\mathcal{H}_{USPS}$  that is contained in a set of hypergraph nodes set are  $\mathcal{NP}$ -hard to approximate within a factor less than  $7/6$  (assuming that  $\mathcal{I}_{USPS}$  or  $\mathcal{H}_{USPS}$  are given by an independence oracle and the weights  $w$  are encoded polynomial in the size of the given path set).

## 5.5 Finding Maximum Unique Shortest Path Systems

The opposite problem to finding an irreducible non-USPS in a given path set  $S \subseteq \mathcal{P}$  is to find a maximum USPS in  $S$ . Again, we are not only interested in finding an inclusion-wise maximum such set, but a maximum cardinality or maximum weight one.

These problems have an interesting direct application in telecommunication network planning: Several routing protocols allow to specify for each commodity independently whether it shall be routed along an individually

pre-configured path or along a default route, which typically is a shortest path with respect to some global routing metric. The most prominent examples of such protocols are MPLS, PNNI, and BGP. The possibility to configure individual routing paths as substitutes for the default routes makes these protocols more flexible and powerful than standard shortest path routing protocols such as OSPF or IS-IS. Network operators however, are very reluctant to set up such individual paths, because this involves additional configuration and management overhead. An interesting problem in such networks is to find a maximum USPS in a set of prescribed end-to-end routing paths, which may be the result of some previous routing optimization process. The paths in the USPS then can be managed by the default shortest path routing mechanism and only the remaining ones need to be configured manually.

The task of finding a maximum cardinality USPS within a given path set can be formalized as the MAX-USPS problem:

<i>Problem:</i>	MAX-USPS
<i>Instance:</i>	A digraph $D = (V, A)$ and a path set $S \subseteq \mathcal{P}$ .
<i>Solution:</i>	A USPS $R \subseteq S$ .
<i>Objective:</i>	$\max  R $ .

The more general task of finding a USPS of maximum weight leads to the following optimization problem:

<i>Problem:</i>	MAX-WEIGHT-USPS
<i>Instance:</i>	A digraph $D = (V, A)$ and non-negative weights $w \in \mathbb{Z}_+^{\mathcal{P}}$ .
<i>Solution:</i>	A USPS $R \subseteq \mathcal{P}$ .
<i>Objective:</i>	$\max \sum_{P \in R} w_P$ .

Again, we assume that the weights are given as a sparse-vector in MAX-WEIGHT-USPS, that is, only those weights with  $w_P > 0$  are actually encoded in a MAX-WEIGHT-USPS instance.

Clearly, both MAX-USPS and MAX-WEIGHT-USPS are  $\mathcal{NPO}$  problems.

To unify the notation in the following, we let  $S := \{P \in \mathcal{P} : w_P > 0\}$  for any given instance  $(D, w)$  of MAX-WEIGHT-USPS. Clearly, each instance  $(D, w)$  of MAX-WEIGHT-USPS has an optimal solution  $R$  with  $R \subseteq S$ . (Note that we do not require that  $R$  is maximal with respect to set-inclusion in the entire path set  $\mathcal{P}$ .) So, in both problems we are looking for a USPS  $R$  within the (implicitly) given path set  $S$ .

### 5.5.1 Greedy Approaches

As the family of all unique shortest path systems in the given digraph forms an independence system, the most natural approach for computing solutions

of MAX-USPS and MAX-WEIGHT-USPS is to use the standard primal or dual greedy algorithm for finding a maximum cardinality or a maximum weight independent set in a general independence system, see Edmonds [77] or Korte et al. [126] for example.

Starting with  $R := \emptyset$ , the primal greedy algorithm iteratively tries to add the paths  $P \in S$  (in order of decreasing weight) to  $R$ . If the augmented path set  $R \cup P$  is a USPS, the path  $P$  is added permanently to  $R$  and the algorithm continues with the next path in  $R$ . Otherwise the primal greedy algorithm proceeds with the next path in  $R$  without adding  $P$ .

The dual greedy algorithm, on the other hand, starts with  $R := S$  and iteratively removes a (minimum weight) path from  $R$  until the remaining path set becomes a USPS.

Clearly, both algorithms run polynomial in the problem's encoding size. Each executes at most  $|S|$  many iterations and, in each iteration, the test whether the current path set is a USPS or not can be performed in polynomial time using the linear programming techniques discussed in Section 4.3.

As the constructed USPS  $R \subseteq S$  will contain at least one path which, in the weighted case, is a maximum weight path, both algorithms achieve a worst-case performance ratio of  $|S|$ .

To see that this worst-case performance guarantee is tight, we consider the rank quotient [118, 113, 125] of the underlying independence system  $\mathcal{I}_{USPS}$ . For each path set  $S$ , we denote its **lower rank** by

$$lr(S) := \min\{|R| : R \subseteq S \text{ is a USPS and} \\ \text{there is no USPS } R' \text{ with } R \subsetneq R' \subseteq S\}$$

and its **upper rank** by

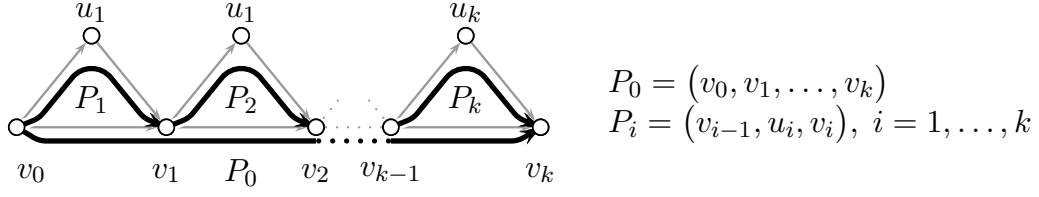
$$ur(S) := \max\{|R| : R \subseteq S \text{ is a USPS}\}, \text{ respectively.}$$

The **rank quotient** for the family of shortest path systems then is

$$q(\mathcal{I}_{USPS}) := \min\{lr(S)/ur(S) : S \subseteq \mathcal{P} \text{ and } ur(S) \neq 0\}.$$

It was shown by Jenkyns et al. [118, 113, 125] that the inverse rank quotient is a tight bound for the worst case approximation ratio achieved by both the primal and the dual greedy algorithm for the problem of finding a maximum weight independent set in a general independence system.

For an independence system that is the intersection of finitely many matroids, the rank quotient is bounded from below by a constant that depends on the number of intersecting matroids. Unfortunately, the independence system  $\mathcal{I}_{USPS}$  of unique shortest path systems cannot be described as an intersection of finitely many matroids in general. The following example shows that the quotient between the lower and the upper rank can be as small as  $1/(|S| + 1)$  for a path set  $S \subseteq \mathcal{P}$ .



**Figure 5.9** Path set with rank quotient  $1/k$ .

**Example 5.3** Let  $k \in \mathbb{Z}_+$ . Consider the digraph  $D_k = (V_k, A_k)$  given by

$$V_k := \{v_0, \dots, v_k, u_1, \dots, u_k\} \text{ and}$$

$$A_k := \{(v_{i-1}, v_i), (v_{i-1}, u_i), (u_i, v_i) : i = 1, \dots, k\}$$

and the path set  $S_k = \{P_0, \dots, P_k\}$  consisting of the path

$$P_0 := (v_0, v_1, \dots, v_k) \text{ and}$$

$$P_i := (v_{i-1}, u_i, v_i) \text{ for } i = 1, \dots, k.$$

Figure 5.9 illustrates  $D_k$  and  $S_k$ .

For any  $i = 1, \dots, k$ , the paths  $P_0$  and  $P_i$  contain different  $(v_{i-1}, v_i)$ -(sub)paths. Thus, either  $P_0$  or  $P_i$  but not both can be uniquely determined shortest paths between their respective terminals for the same metric  $\lambda$ . One easily verifies that the pairs  $\{P_0, P_i\}$  are the only irreducible non-USPSs contained in  $S_k$  and that the path set  $\{P_i : i = 1, \dots, k\}$  as well as the path set  $\{P_0\}$  are (inclusion-wise) maximal USPSs contained in  $S_k$ . Hence,  $S_k$  has a rank quotient of (at most)  $1/k$ .  $\triangle$

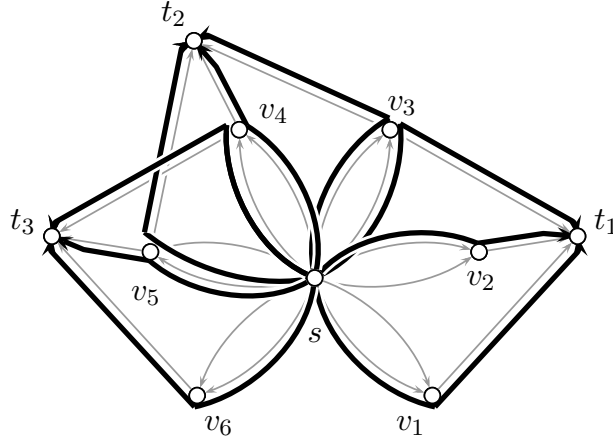
It now follows immediately from the results of Jenkyns et al. [118, 113, 125] that for both the primal and the dual greedy algorithm the worst-case approximation guarantee of  $|S|$  is (asymptotically) tight for both problems MAX-USPS and MAX-WEIGHT-USPS.

In the remainder of this section we show that it is hard to approximate these two problems within a small constant factor.

### 5.5.2 Inapproximability Results

If the path set  $S$  given as input in the problems MAX-USPS or MAX-WEIGHT-USPS is a USPS, then  $S$  itself is an optimal solution. If  $S$  is an irreducible non-USPS, then one of the  $|S|$  many subsets  $R \subset S$  with  $|R| = |S| - 1$  is a optimal solution. With the methods presented in Chapter 4 both cases can be recognized and solved in polynomial time.

In general, however, the problems MAX-USPS and MAX-WEIGHT-USPS are not only hard to solve to optimality, but also hard to approximate within a constant factor less than  $8/7$ .



**Figure 5.10** Path set  $S$  in  $D$  corresponding to the clauses  $C_1 = (\bar{x}_1 \vee x_2 \vee \bar{x}_3)$ ,  $C_2 = (x_3 \vee \bar{x}_4 \vee x_5)$ , and  $C_3 = (x_4 \vee x_5 \vee x_6)$ .

**Theorem 5.17** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate MAX-USPS within a factor of  $8/7 - \epsilon$ . This holds even if each path  $P \in S$  is a shortest  $(s_P, t_P)$ -path (w.r.t. the number of arcs) with  $|P| = 2$ .*

**Proof.** We construct an approximation preserving reduction from the optimization problem MAX-3-SAT to MAX-USPS. The problem MAX-3-SAT is defined as follows: Given a set  $X$  of boolean variables, a collection  $C$  of disjunctive clauses of at most three literals (i.e., a negated or unnegated variable) per clause, find a truth assignment for  $X$  that maximizes the number of clauses evaluating to *true*. This problem was shown to be  $\mathcal{APX}$ -complete by Papadimitriou and Yannakakis [156]. Håstad [112] proved that it is  $\mathcal{NP}$ -hard to approximate MAX-3-SAT within a factor strictly less than  $8/7$ .

Suppose we are given an instance of MAX-3-SAT with variables  $x_1, \dots, x_q$  and clauses  $C_1, \dots, C_r$ . W.l.o.g., we may assume that no clause contains the same boolean variable twice. We construct a MAX-USPS instance consisting of a digraph  $D = (V, A)$  and a path set  $S \subseteq \mathcal{P}_D$  as shown in Figure 5.10.

The node set  $V$  consist of one center node  $s$ , the nodes  $v_i$  for all  $i \in \{1, \dots, q\}$ , and the nodes  $t_l$  for all  $l \in \{1, \dots, r\}$ . The arcs set  $A$  consists of two parallel arcs  $a_i$  and  $\bar{a}_i$  from  $s$  to  $v_i$  for each  $i \in \{1, \dots, q\}$ , and of one arc  $(v_i, t_l)$  for all  $i \in \{1, \dots, q\}$  and  $l \in \{1, \dots, r\}$ .

The path set  $S$  contains one path for each occurrence of a boolean variable in a clause. If the boolean variable  $x_i$  occurs in clause  $C_l$  unnegated, i.e.,  $C_l = (x_i \vee \dots)$ , then  $S$  contains the path

$$P_{l,i} := (s, a_i, v_i(v_i, t_l), t_l) .$$

If  $x_i$  occurs in  $C_l$  negated, i.e.,  $C_l = (\bar{x}_i \vee \dots)$ , then  $S$  contains the path

$$\bar{P}_{l,i} := (s, \bar{a}_i, v_i(v_i, t_l), t_l) .$$



In the following, we show that each truth assignment for the boolean variables that satisfies  $c$  clauses corresponds to a USPS  $R \subseteq S$  with  $|R| = c$  and, in the other direction, any USPS  $R \subseteq S$  can be transformed into a truth assignment  $x$  that satisfies at least  $|R|$  many clauses.

First, consider a truth assignment  $x$  which satisfies  $c$  clauses. W.l.o.g., we may assume that these are the clauses  $C_1$  through  $C_c$ . We define the corresponding path set  $R = R(x) \subseteq \mathcal{P}$  as follows: For each clause  $C_l$  with  $l \in \{1, \dots, c\}$ , at least one of its literals must evaluate to *true*. (If two or more literals evaluate to *true*, we choose the lexicographically first one.) If this literal is an unnegated variable  $x_i$ , then the path  $P_{l,i}$  belongs to  $R$ . Otherwise, if the literal evaluating to *true* in  $C_l$  is a negated variable  $\bar{x}_i$ , then  $R$  contains the path  $\bar{P}_{l,i}$ . For clauses  $C_l$  with  $l \geq c+1$ , there is no path in  $R$ .

Clearly,  $R \subseteq S$  and  $|R| = c$ .

To see that  $R$  is a USPS, note that the same value *true* or *false* is assigned to all occurrences of a boolean variable  $x_i$ . Thus, the path set  $R$  cannot contain two path  $P$  and  $\bar{P}$  with  $a_i \in P$  and  $\bar{a}_i \in \bar{P}$ . Furthermore,  $R$  contains exactly one  $(s, t_l)$ -path for each  $l \in \{1, \dots, c\}$ . Consequently, the paths in  $R$  form an arborescence with root  $s$ , which implies that  $R$  is a USPS.

Now, secondly, let  $R \subseteq S$  be a USPS with  $|R| = c$ . We construct a truth assignment  $x := x(R) \in \{\text{true}, \text{false}\}^q$  for the boolean variables as follows: If  $P_{l,i} \in R$  for some  $l \in \{1, \dots, r\}$  and  $i \in \{1, \dots, q\}$ , then we set  $x_i = \text{true}$ . If  $\bar{P}_{l,i} \in R$ , we assign  $x_i = \text{false}$ . All boolean variables that are not determined by this procedure are set to *true*.

Since  $R$  is a USPS, it contains no pairs of conflicting paths. In particular, either arc  $a_i$  or  $\bar{a}_i$  or none of both is contained in the paths of  $S$ . This implies that there are no two paths  $P_{k,i}$  and  $\bar{P}_{l,i}$  in  $S$  and, therefore, the truth assignment  $x$  is well-defined.

Furthermore,  $x$  satisfies at least all those clauses  $C_l$  where  $R$  contains an  $(s, t_l)$ -path. Because  $R$  is a USPS, it contains at most one  $(s, t_l)$ -path for each  $l = 1, \dots, r$ , which implies that the truth assignment  $x$  satisfies at least  $|R| = c$  many clauses.

Clearly, the construction of  $D$  and  $S$  as well as the construction of  $x = x(R)$  from  $R$  are polynomial in the size of the given MAX-3-SAT instance. Therefore, any polynomial time approximation algorithm for MAX-USPS carries over to a polynomial time approximation algorithm for MAX-3-SAT with the same approximation ratio. Consequently, MAX-USPS cannot be approximated within a factor less than  $8/7$ , unless  $\mathcal{P} = \mathcal{NP}$ .  $\square$

Note that the optimal solution value of MAX-USPS is exactly the rank of the given path set  $S$ . Thus, computing a close approximation of the rank of a non-USPS is  $\mathcal{NP}$ -hard in general.

**Corollary 5.18** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate the rank of an arbitrary path set within a factor of  $8/7 - \epsilon$ .*

Theorem 5.17 carries over straightforward to the more general, weighted problem version MAX-WEIGHT-USPS.

**Corollary 5.19** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate the problem MAX-WEIGHT-USPS within a factor of  $8/7 - \epsilon$ .*

Again, it is not difficult to verify that Corollary 5.19 also holds if the path weights are given implicitly as an additive function of some arc weights  $c \in \mathbb{Z}_+^A$ , i.e.,  $w_P := \sum_{a \in P} c_a$  for each path  $P \in \mathcal{P}$ .

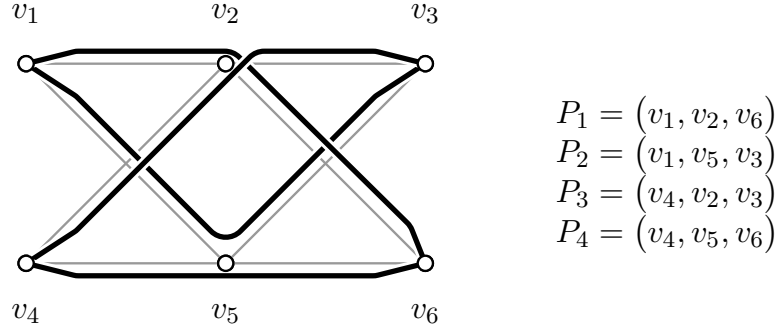
**Remark 5.20** The only particular feature of unique shortest path systems used in the proof of Theorem 5.17 is the Bellman property. Therefore, Theorem 5.17 and its corollaries hold also for the problem of finding a maximum cardinality or a maximum weight path set  $R \subseteq S$  that satisfies only the Bellman property, but which is not necessarily a USPS. In particular, Theorem 5.17 holds for the problem of finding a maximum cardinality or a maximum weight so-called *unsplittable source invariant routing* within a given path set, see also Chapter 6.

In the language of independence systems and stable sets Corollary 5.19 states that the problem of finding a maximum weight independent set in  $(\mathcal{P}, \mathcal{I}_{USPS})$  and, equivalently, the problem of finding a maximum weight stable set in  $\mathcal{H}_{USPS}$  are  $\mathcal{NP}$ -hard to approximate within a factor strictly less than  $8/7$  (provided that the independence system  $\mathcal{I}_{USPS}$  or the hypergraph  $\mathcal{H}_{USPS}$  are given implicitly by an independence oracle and the encoding size of the weights  $w$  is polynomial in the size of the underlying digraph).

## 5.6 Undirected Unique Shortest Paths Systems

Although traffic demands and flows are typically asymmetric in real networks, also undirected unique shortest path systems are of practical relevance. For operational reasons network administrators often require that the routing lengths for both directions of a bidirectional link are equal. In this case, the routing can be modeled by path systems in an undirected graph, which reduces the size and the complexity of the models enormously.

The definitions, properties, algorithms, and complexity results discussed above carry over immediately to the undirected case. Given an undirected graph  $G = (V, E)$  and letting  $\mathcal{P}$  denote the set of all undirected simple paths in  $G$ , a metric  $\lambda \in \mathbb{R}_+^E$  is said to be compatible with a given path set  $S \subseteq \mathcal{P}$  if each path  $P \in S$  is the uniquely determined shortest  $(s_P, t_P)$ -path with respect to  $\lambda$ . A path set  $S \subseteq \mathcal{P}$  is an undirected USPS if there exists a compatible metric  $\lambda \in \mathbb{R}_+^E$  for  $S$ .



**Figure 5.11** Undirected non-USPS with B-property.

As in the directed case, any metric that is compatible with a given path set can be transformed to an integer-valued one and any undirected USPS can be extended to an USPS containing an  $(s, t)$ -path for node pair  $(s, t) \in V^{(2)}$  where  $s$  and  $t$  are connected in  $G$ . The B-property, the GB-property, and the valid-cycle property are fulfilled by any undirected USPS, but none of these properties is sufficient to completely characterize all undirected USPS in general. An example of an undirected path set that has the B-property but is no USPS is shown in Figure 5.11.

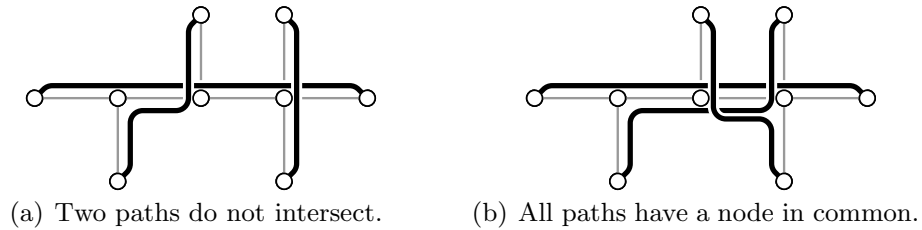
Undirected USPS containing three or less paths are completely characterized by the B-property, too. Yet, the proof of this fact is more complicated than in the directed case.

**Theorem 5.21** *An undirected path set  $S \subseteq \mathcal{P}$  with  $|S| \leq 3$  is an USPS if and only if  $S$  has the B-property.*

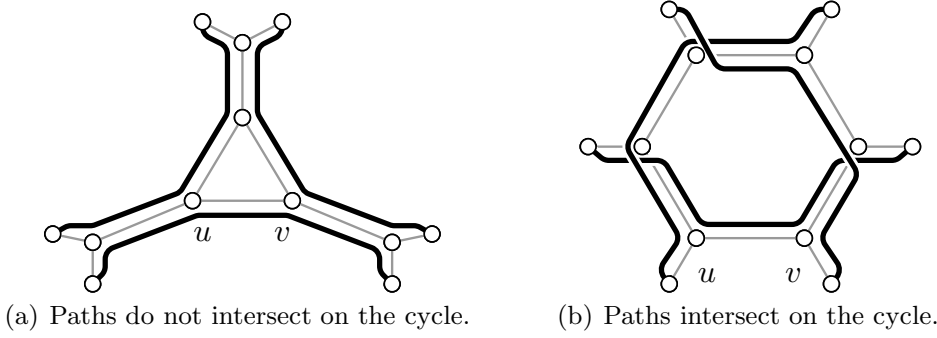
**Proof.** The cases  $|S| = 1$  and  $|S| = 2$  follow analogous to the directed variant in Theorem 5.7. It remains to prove Theorem 5.21 for  $|S| = 3$ .

Let  $S = \{P_1, P_2, P_3\}$  be a path set in the undirected graph  $G = (V, E)$  such that  $S$  has the B-property. Define  $G' := (V, E')$  with  $E' := \{e \in E : e \in P_1 \cup P_2 \cup P_3\}$ . If two of the three paths do not intersect or all three paths have one node in common, then  $G'$  is a forest, see Figure 5.12. In this case, the claim follows immediately.

Otherwise all three paths intersect pairwise but share no common node. Then  $G'$  contains exactly one cycle, see Figure 5.13. Furthermore, this cycle contains at least one edge, say  $f = uv$ , that is contained in only one of the paths, say  $P_3$ . As  $G' - f$  is a tree, there is exactly one path between  $u$  and



**Figure 5.12** Undirected path sets of size three with B-property that form a forest.



**Figure 5.13** Undirected path sets of size three with B-property where all paths intersect pairwise.

$v$  in  $G' - f$ . We denote the distance between  $u$  and  $v$  in  $G' - f$  by  $d(u, v)$ . Then

$$\lambda_e := \begin{cases} 1 & , \text{ if } e \in P_1 \cup P_2 \cup P_3 \setminus \{f\}, \\ d(u, v) - \frac{1}{2} & , \text{ if } e = f, \text{ and} \\ |V| & , \text{ otherwise,} \end{cases}$$

is a compatible edge length function for  $S$ . □

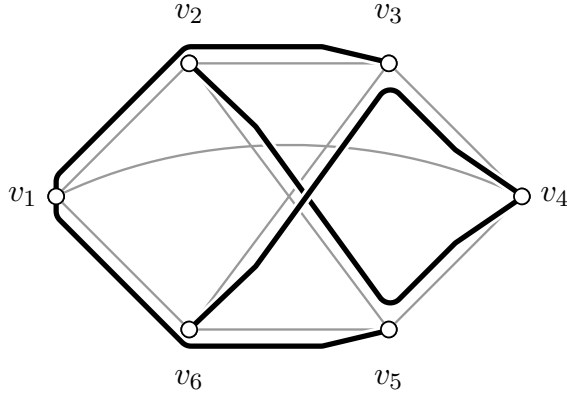
For undirected path sets, Ben-Ameur and Gourdin [27] introduced the properties *cyclic comparability* and *generalized cyclic comparability*. The generalized cyclic comparability property is equivalent to the GB-property. The cyclic comparability property corresponds to a weaker variant of the GB-property, where only the removal of single edges is considered. For the sake of completeness, we review the definition of cyclic comparability and its relation to the other properties here.

**Definition 5.22 (Ben-Ameur and Gourdin [27])** Let  $G = (V, E)$  be an undirected graph. A path set  $S \subseteq \mathcal{P}$  has the **cyclic comparability property** if, for every  $f \in E$  that is not a bridge of  $G$ , there exists a cycle  $C$  in  $G$  with  $f \in C$  such that either  $P[u, v] = \emptyset$  or  $P[u, v] \subset C$  holds for all pairs of distinct nodes  $u, v \in C$  and for all paths  $P \in S$ .

**Theorem 5.23 (Ben-Ameur and Gourdin [27])** Any undirected USPS has the B-property.

**Observation 5.24 (Ben-Ameur and Gourdin [27])**

- (i) If an undirected path set has the GB-property, then it also has the cyclic comparability property.
- (ii) If an undirected path set has the cyclic comparability property, then it also has the B-property.



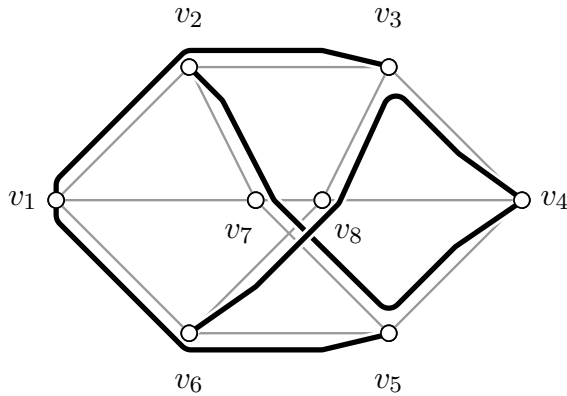
$$\begin{aligned} P_1 &= (v_1, v_2, v_3) \\ P_2 &= (v_1, v_6, v_5) \\ P_3 &= (v_2, v_5, v_4) \\ P_4 &= (v_6, v_3, v_4) \end{aligned}$$

The cyclic comparability is violated for  $F = \{(1, 4)\}$ .

**Figure 5.14** Undirected path set with B-property, but without cyclic comparability property.

In general, the cyclic comparability property is stronger than the B-property and weaker than the GB-property. Figure 5.14 shows an undirected path set with the B-property but without the cyclic comparability property and Figure 5.15 shows a path set that with the cyclic comparability property but without the GB-property.

The linear programming techniques presented in Chapter 4 can be applied analogously to find a compatible metric  $\lambda \in \mathbb{R}_+^E$  for an undirected path set or to prove that none exists. The greedy algorithms for finding an irreducible non-USPS or a maximum USPS within a given path set carry over to the undirected problem versions and achieve the same approximation ratios as in the directed case. The inapproximability results presented in Sections 5.4.2 and 5.5 for MIN-NON-USPS, MIN-WEIGHT-NON-USPS, MAX-USPS, and MAX-WEIGHT-USPS hold for the corresponding undirected problem versions as well. For MAX-USPS and MAX-WEIGHT-USPS the proof carries over literally to the undirected case, while for MIN-NON-USPS and MIN-WEIGHT-NON-USPS a similar construction yields the same inapproxima-



$$\begin{aligned} P_1 &= (v_1, v_2, v_3) \\ P_2 &= (v_1, v_6, v_5) \\ P_3 &= (v_2, v_7, v_5, v_4) \\ P_4 &= (v_6, v_8, v_3, v_4) \end{aligned}$$

The generalized cyclic comparability is violated for  $F = \{(1, 7), (7, 8), (8, 4)\}$ .

**Figure 5.15** Undirected path set with cyclic comparability but without GB-property.

bility threshold.

As in the directed case, the structural properties of undirected unique shortest path systems may become rather complicated. For undirected cycles and some related graphs, however, the family of all undirected USPSs can be easily characterized.

**Definition 5.25** *A **hat-cycle** is an undirected graph  $G = (V_1 \cup V_2, E_1 \cup E_2)$  consisting of an elementary cycle  $C = (V_1, E_1)$  plus some additional nodes  $V_2$  and edges  $E_2$ , such that*

- (i) *each extra node  $u \in V_2$  is adjacent to exactly two nodes  $v_1, v_2 \in V_1$  with  $v_1 v_2 \in E_1$ , and*
- (ii) *for each edge  $v_1 v_2 \in E_1$ , there is only one node  $u \in V_2$  that is adjacent to both  $v_1$  and  $v_2$ .*

It was shown by Ben-Ameur and Gourdin [27] that, in a hat-cycle graph, each path set with the B-property is an USPS. This result can be generalized as follows.

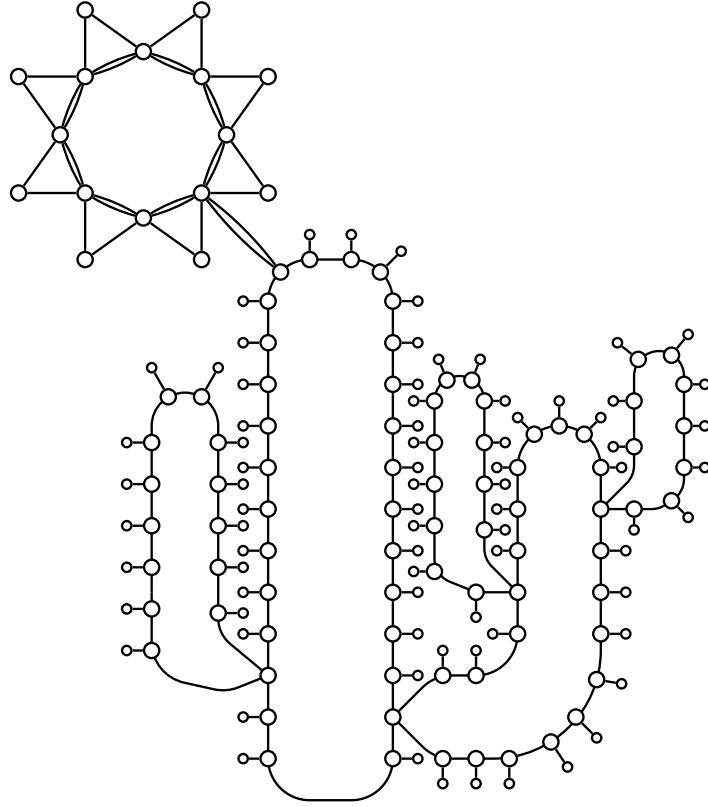
**Theorem 5.26 (Ben-Ameur and Gourdin [27])** *Let  $G = (V, E)$  be an undirected graph and  $G'$  its associated simple graph where all loops are removed and all sets of parallel edges are replaced by a single edge. If each block of  $G'$  is a cycle or a hat-cycle, then each path set  $S \subset \mathcal{P}$  with the B-property is an USPS.*

**Proof.** For the case where  $G$  itself is a cycle or a hat-cycle graph, the claim was shown by Ben-Ameur and Gourdin [27]. From this, it follows immediately that the B-property sufficiently describes the family of all USPS if all blocks of  $G$  are cycle or hat-cycle graphs.

Loops clearly cannot be contained in any simple path. Furthermore, if  $G$  has parallel edges  $e_1, \dots, e_p$  between two of its nodes, then any two paths  $P_1$  and  $P_2$  such that w.l.o.g.  $e_1 \in P_1$  and  $e_2 \in P_2$  are conflicting. Hence, at most one of the parallel edges  $e_1, \dots, e_p$  can be contained in the paths of a path set that has the B-property. This yields the claim.  $\square$

The class of undirected graphs whose associated simple graphs' blocks are cycle or hat-cycle graphs generalizes several other graph classes, for example cactus graphs. Figure 5.16 shows an example of a graph where the B-property sufficiently characterizes all USPS.

**Remark 5.27** Note that there is no directed correspondence of Theorem 5.26. The B-property is not even sufficient to characterize all directed USPSs in a bidirected ring. Figure 5.4 shows an example of a non-USPS with the B-property in a bidirected ring.



**Figure 5.16** In this graph, the B-property completely characterizes all USPSs.

If the B-property is sufficient to characterize the family of all USPS, like it is the case for the undirected graphs described in Theorem 5.26, then each irreducible non-USPS contains exactly two conflicting paths and the conflict hypergraph  $\mathcal{H}_{USPS}$  is a simple graph. In this case, the MIN-NON-USPS and MIN-WEIGHT-NON-USPS problem are trivially solvable.

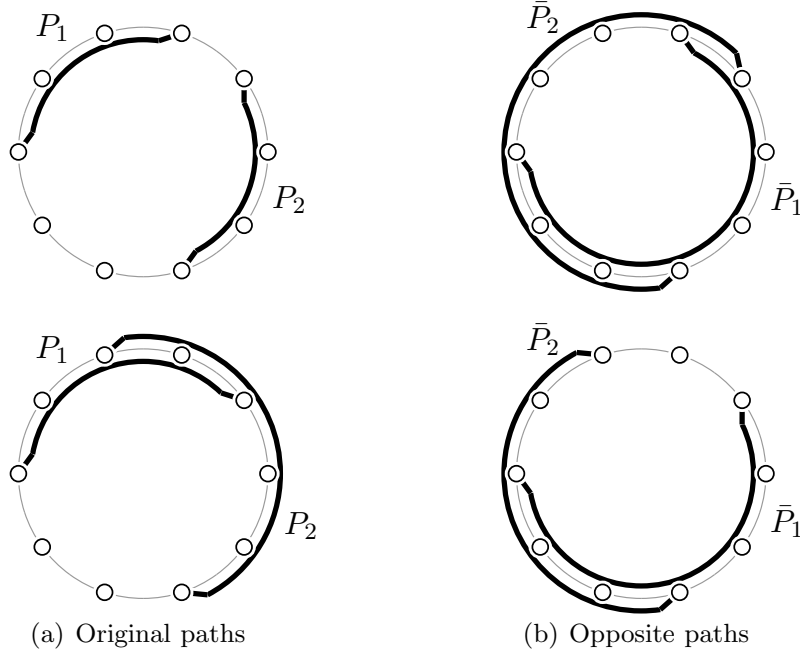
**Theorem 5.28** *The problems MIN-NON-USPS and MIN-WEIGHT-NON-USPS are polynomially solvable if the underlying graph  $G$  belongs to the class of undirected graphs described in Theorem 5.26.*

**Proof.** The only irreducible non-USPSs in  $G$  are pairs of conflicting paths. These can be enumerated in polynomial time.  $\square$

For the case where the underlying graph  $G$  is an undirected cycle, also the problems MAX-USPS and MAX-WEIGHT-USPS become polynomially solvable.

**Theorem 5.29** *The problems MAX-USPS and MAX-WEIGHT-USPS are polynomially solvable if the underlying graph  $G$  is an undirected cycle.*

**Proof.** It is sufficient to show that MAX-WEIGHT-USPS is polynomially solvable.



**Figure 5.17** Opposing paths in an undirected cycle.

Let  $G$  be an undirected cycle and  $w \in \mathbb{Z}_+^{\mathcal{P}}$  be a non-negative weight function on the paths in  $G$ . W.l.o.g., we may assume that  $w_P \geq 0$  for each  $P$ . Since each subset of an USPS is an USPS again, no path  $P$  with  $w_P < 0$  will be contained in an optimal solution.

Clearly, for each path  $P \in \mathcal{P}$ , there is a unique opposite path  $\bar{P}$  between  $s_P$  and  $t_P$  in  $G$  with  $\bar{P} \neq P$ . For each path system  $S$  in  $G$ , we denote by  $\bar{S} := \{\bar{P} : P \in S\}$  the system of opposite paths. Furthermore, let  $\bar{w} \in \mathbb{Z}_+^{\mathcal{P}}$  be the path weights defined by  $\bar{w}_P := w_{\bar{P}}$ .

By Theorem 5.26, the B-property sufficiently describes all USPS in  $G$ . Hence, the problem of finding a maximum weight USPS and the problem of finding a maximum weight path system that contains no pairs of conflicting paths are equivalent.

Now, consider the circular arc graph  $\mathcal{H}_{CA}$  defined by all paths  $\mathcal{P}$  in  $G$ . The nodes of this graph are the paths in  $G$  and there is an edge  $P_1 P_2$  in  $\mathcal{H}_{CA}$  if and only if the paths  $P_1$  and  $P_2$  intersect in at least one arc. Two paths  $P_1$  and  $P_2$  conflict if and only if they cover the entire cycle  $G$ . And this is the case if and only if their complements  $\bar{P}_1$  and  $\bar{P}_2$  are arc disjoint, see Figure 5.17. In other words, there is an edge  $\bar{P}_1 \bar{P}_2$  in  $\mathcal{H}_{CA}$  if and only if  $P_1$  and  $P_2$  do not conflict with each other. Consequently, each clique  $\bar{S}$  in  $\mathcal{H}_{CA}$  is the opposite path system of the path system  $S$  without conflicting paths. Finding a maximum weight unique shortest path system for the weights  $w$  therefore is equivalent to finding a maximum weight clique in  $\mathcal{H}_{CA}$  with respect to the opposite weights  $\bar{w}$ .

The MAXIMUM WEIGHT CLIQUE problem can be solved polynomially for circular arc graphs with the algorithms of Gavril [97] or Wilfong and Win-



kler [191], for example. Hence, also MAX-WEIGHT-USPS is polynomially solvable in undirected cycles.  $\square$

Note that Theorem 5.29 does not carry over to the larger class of undirected graphs described in Theorem 5.26. Even though  $\mathcal{H}_{USPS}$  is a simple graph, it may contain non-trivial substructures such as odd holes. It is not difficult to show that MAX-USPS and MAX-WEIGHT-USPS on cactus graphs are at least as hard to approximate as the MAXIMUM CUT problem.

## 5.7 Unique Shortest Path Forwardings

In this section, we finally consider the forwarding representation of unsplit-table shortest path routings where, instead of the end-to-end routing paths themselves, we are given a set of forwarding arcs that describe via which arcs the traffic streams must leave the nodes.

The basic notions for this routing representation have been introduced in Chapter 4, page 60ff. A *forwarding* is a set  $F \subseteq V \times A$ . Each *forwarding arc*  $(t, (u, v) \in F) \in F$  means that the traffic from or via node  $u$  towards destination  $t$  must leave node  $u$  via the arc  $(u, v)$ . We say that

- (i) a metric  $\lambda \in \mathbb{R}_+^A$  is *compatible* with a forwarding  $F \subseteq V \times A$  if, for each  $(t, (u, v)) \in F$ , arc  $(u, v)$  is contained in *all* shortest  $(u, t)$ -paths with respect to  $\lambda$ , and
- (ii) a forwarding  $F \subseteq V \times A$  is a *unique shortest path forwarding (USPF)* if there exists a compatible metric for  $F$ . Otherwise  $F$  is called a *non-USPF*.

Note that a forwarding not necessarily prescribes complete end-to-end routing paths, but only some arcs that must be contained in these paths. Accordingly, a compatible metric need not induce unique shortest end-to-end paths, it only must ensure that the given forwarding arcs are uniquely induced. A forwarding therefore is a USPF if it either corresponds to a unique shortest path routing or if it can be extended to a unique shortest path routing, for whichever commodity set.

As for the path set representation, we are interested in the combinatorial properties of unique shortest path forwardings and in the problems of finding a minimal non-USPF or a maximal USPF within a given forwarding.

Clearly, any subset of an USPF is an USPF as well. Thus, the family of all USPF in  $D$  forms an independence system. The circuits of this independence system are the **irreducible non-USPFs**. We denote the families of all USPFs and of all irreducible non-USPFs by

$$\begin{aligned} \mathcal{I}_{USPF} &:= \{ F \subseteq V \times A : F \text{ is an USPF} \} \subseteq 2^{V \times A} \quad \text{and} \\ \mathcal{C}_{USPF} &:= \{ F \subseteq V \times A : F \text{ is an irreducible non-USPF} \} \subseteq 2^{V \times A}, \end{aligned}$$

respectively.

Most of the properties, algorithms, and complexity results discussed in the previous sections for unique shortest path system carry over directly to unique shortest path forwardings.

As we have already shown in Chapter 4, the INVERSE UNIQUE SHORTEST PATH FORWARDING problem (IUSPF) of finding a compatible metric for a given forwarding  $F \subseteq V \times A$  or proving that no such metric exists can be solved in polynomial time. Thus, one can decide in polynomial time whether a given forwarding  $F$  is a USPF or not.

Using the same techniques and examples as for the path-set representation, one immediately finds that all inclusion-wise maximal USPFs have the same cardinality and that the rank quotient of the independence system  $(V \times A, \mathcal{I}_{USPF})$  may become arbitrarily small. Also, the Bellman-property, the generalized Bellman property, and the valid-cycle property of Broström and Holmberg [46] can be easily formulated in terms of forwarding arcs. As in the path-case, the Bellman property yields a complete characterization of unique shortest path forwardings in undirected graphs that belong to the graph class described in Theorem 5.26, but none of the aforementioned properties is sufficient in the general case.

The problem of finding an inclusion-wise maximum USPF within a given forwarding can be solved in polynomial time with a standard greedy algorithm that uses the linear programming techniques discussed in Chapter 4 to check if the intermediate candidate forwardings are USPFs or not. On the other hand, the maximum cardinality or the maximum weight of such a USPF cannot be approximated within a factor less than  $8/7$ , unless  $\mathcal{P} = \mathcal{NP}$ . The proof of Theorem 5.17 literally carries over to the forwarding problem version by replacing each path  $P = (v_0, a_1, v_1, \dots, v_k)$  by its first forwarding arc  $(v_0, a_1)$ .

The only interesting problems are the problem of finding some irreducible non-USPF within a given forwarding and the problems of finding a minimum cardinality or a minimum weight such non-USPF. Also for these problems we obtain exactly the same results as for the corresponding path-set problems. The algorithms and proofs are very similar to those for the corresponding path-set problems, but they do not carry over as easily as those for the opposite maximum USPF problems. In the following, we therefore discuss these problems in more detail.

Formally, the two problems of finding a minimum cardinality or a minimum weight non-USPF within a given forwarding are defined as follows.

<i>Problem:</i>	MIN-NON-USPF
<i>Instance:</i>	A digraph $D = (V, A)$ and a non-USPF $F \subseteq V \times A$ .
<i>Solution:</i>	An irreducible non-USPF $E \subseteq F$ .
<i>Objective:</i>	$\min  E $ .

<i>Problem:</i>	MIN-WEIGHT-NON-USPF
<i>Instance:</i>	A digraph $D = (V, A)$ and strictly positive weights $w \in \mathbb{Z}_+^{V \times A}$ .
<i>Solution:</i>	An irreducible non-USPF $E \subseteq V \times A$ .
<i>Objective:</i>	$\min \sum_{(t,a) \in E} w_{t,a}$ .

Because the size of a forwarding is polynomial in the size of the underlying digraph anyway, we do not need to restrict the domain of the weight function to a subset of  $V \times A$  in MIN-WEIGHT-NON-USPF like in the path-set version. To ensure that the optimal solution value is strictly positive, we require that the given forwarding in MIN-NON-USPF is a non-USPF and that the weights in MIN-WEIGHT-NON-USPF are strictly positive.

The simplest solution approach to these problems is to greedily remove forwarding arcs. Starting with the given forwarding  $E := F$ , or with  $E := V \times A$  in the weighted case, we iteratively tries to remove each forwarding arc from  $E$ . If the remaining forwarding still comprises a non-USPF, the forwarding arc is removed permanently and the algorithm continues with the next path. Otherwise we proceeds with the next forwarding arc without removing the current one. Whether or not removing the current forwarding arc would yield a USPF can be checked by solving the corresponding INVERSE UNIQUE SHORTEST PATH FORWARDING (IUSPF) problem. If all forwarding arcs have been considered for removal, the algorithm finishes returning the residual forwarding. In the weighted problem version, the forwarding arcs are considered for removal in order of non-increasing weights.

Obviously, this simple greedy algorithm returns an irreducible non-USPF. Each intermediate IUSPF problem can be solved in polynomial time by solving the corresponding linear system (4.11)–(4.14) discussed in Chapter 4. Hence, the entire algorithm can be implemented to run in polynomial time.

Analogous to the procedure for finding an irreducible non-USPS in a given path set, the practical performance of this algorithm can be improved significantly by reducing the forwarding in each iteration based on the dual Farkas ray that is computed for each infeasible linear system (4.11)–(4.14).

**Observation 5.30** *Let  $F \subseteq V \times A$  be a non-USPF and let  $(\mu, \eta, \dots)$  be a dual Farkas ray for the corresponding linear system (4.11)–(4.14) with  $\mu$  being the dual multipliers for the equalities (4.11) and  $\eta$  being the dual multipliers for the inequalities (4.12). Furthermore, let*

$$\begin{aligned} \text{con}^T(\mu) &:= \{(t, (u, v)) \in F : \mu_{(t, (u, v))} \neq 0\} , \text{ and} \\ \text{con}^N(\eta) &:= \{(t, (u, v)) \in F : \eta_{(t, (u, v'))} \neq 0 \text{ for some } (t, (u, v')) \in \bar{F} \\ &\quad \text{with } v \neq v'.\} . \end{aligned}$$

*Then the set  $\text{con}(\mu, \eta) := \text{con}^T(\mu) \cup \text{con}^N(\eta)$  is a non-USPF.*

The set  $\text{con}^T(\mu)$  contains a set of pairs  $(t, (u, v)) \in F$  where the requirement that arc  $(u, v)$  is on the shortest  $(u, t)$ -path is relevant in the infeasibility proof given by the dual Farkas ray. The other set  $\text{con}^N(\eta)$  contains pairs  $(t, (u, v)) \in F$  where the indirectly implied requirement that some arc  $(u, v')$  with  $v' \neq v$  is *not* in the shortest  $(u, t)$ -path is relevant.

Observation 5.30 leads to the improved dual Greedy algorithm **Greedy-Non-USPF** shown below: Whenever during the dual Greedy algorithm the linear system (4.11)–(4.14) is infeasible for the current forwarding, we take the dual Farkas ray  $\mu$  computed by the LP solver and replace the current forwarding by  $\text{con}(\mu)$ .

**Theorem 5.31** *Algorithm Greedy-Non-USPF computes an irreducible non-USPF  $E \subseteq F$  or states that  $F$  is a USPF in polynomial time.*

Instead of the dual Farkas ray, we could also use irreducible inconsistent subsystems of (4.11)–(4.14) to reduce the size of the candidate forwarding in algorithm **Greedy-Non-USPF**. If  $F \subseteq V \times A$  is a non-USPF and  $I$  is an (irreducible) inconsistent subsystem of (in)equalities of the corresponding linear system (4.11)–(4.14), then the forwarding

$$\begin{aligned} \text{con}(I) := & \{ (t, (u, v)) \in F : I \text{ contains equality (4.11) for } (t, (u, v)) \} \\ & \cup \{ (t, (u, v)) \in F : I \text{ contains inequality (4.12) for some} \\ & \quad (t, (u, v')) \in \bar{F} \text{ with } v \neq v' \} . \end{aligned}$$

is a non-USPF. However, note that  $\text{con}(I)$  is not necessarily irreducible, even if  $I$  is an irreducible inconsistent subsystem.

One easily observes that algorithm **Greedy-Non-USPF** achieves a worst case performance ratio of  $|F|$  for MIN-NON-USPF and of  $|V| \cdot |A|$  for MIN-

---

**Algorithm 5.2** Greedy-Non-USPF

---

**Input:** MIN-NON-USPF: Forwarding  $F \subseteq V \times A$

(**Input Min-Weight-Non-USPF:** Weights  $w \in \mathbb{Z}_+^{V \times A}$  and  $F := V \times A$ )

**Output:** Irreducible non-USPF  $E \subseteq F$ .

If (4.11)–(4.14) is feasible for  $F$  then

    Return ' $F$  is a USPF'.

Let  $(\mu, \eta, \dots)$  be a dual Farkas ray of (4.11)–(4.14) for  $F$ .

Set  $E := \text{con}(\mu, \eta)$ .

For each  $(t, a) \in E$  do (in order of non-increasing weights  $w_{(t, a)}$ )

    If (4.11)–(4.14) is infeasible for  $E \setminus \{(t, a)\}$  then

        Let  $(\mu, \eta, \dots)$  be a corresponding dual Farkas ray.

        Set  $E := \text{con}(\mu, \eta)$ .

Return  $E$ .

---

WEIGHT-NON-USPF. It is also not difficult to construct examples where ratios of  $\Theta(|F|)$  and  $\Theta(|V| \cdot |A|)$  are attained.

In the remainder of this section, we prove that it is computationally hard to approximate MIN-NON-USPF or MIN-WEIGHT-NON-USPF within a factor strictly less than  $7/6$ .

**Theorem 5.32** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate MIN-NON-USPF within a factor of  $7/6 - \epsilon$ .*

**Proof.** Except for some minor details, the proof is analogous to the proof of Theorem 5.15. For this reason, we only discuss the differences in the problem reduction and in the main proof steps here.

Again, let  $H$  be a given instance of MINIMUM VERTEX COVER with nodes  $w_i$ ,  $i \in I := \{1, \dots, n\}$ , and edges  $f_k$ ,  $k \in K := \{1, \dots, m\}$ , and let  $\alpha \in \mathbb{Z}_+$  be a sufficiently large integer number.

We construct a digraph  $D = (V, A)$  and a path set  $S \subseteq \mathcal{P}_D$  exactly as in the proof of Theorem 5.15. Recall that  $S$  consists of four different types of paths, i.e.,

$$S := \bigcup_{i \in I} S_i^1 \cup \bigcup_{k \in K} S_k^2 \cup \bigcup_{i \in I, k \in K: f_k \in \delta(w_i)} S_{i,k}^3 \cup S_{i,k}^4 .$$

To define an instance of MIN-NON-USPF, we now construct a forwarding  $F \subseteq V \times A$  that corresponds to the paths in  $S$  but which is only a subset of  $\mathcal{F}(S)$ .

For each path  $P \in \bigcup S_i^1$ , we only add the  $(t_P, a)$  pair corresponding to the first arc  $a \in P$  to  $F$ . For each other path  $P \in \bigcup S_k^2 \cup \bigcup S_{i,k}^3 \cup \bigcup S_{i,k}^4$ , the set  $F$  contains all  $(t_P, a)$  pairs with  $a \in P$ . More formally, we let

$$\begin{aligned} F^1 &:= \{ (u_i^{l+1}, (\bar{u}_i^l, u_i^l)), (\bar{u}_i^{l+1}, (u_i^l, \bar{u}_i^l)) : i \in I, l = 0, \dots, \alpha - 1 \} , \\ F^2 &:= \bigcup_{k \in K} \mathcal{F}(S_k^2) \cup \bigcup_{i \in I, k \in K: f_k \in \delta(w_i)} \mathcal{F}(S_{i,k}^3) \cup \mathcal{F}(S_{i,k}^4) \quad , \text{ and} \\ F &:= F^1 \cup F^2 . \end{aligned}$$

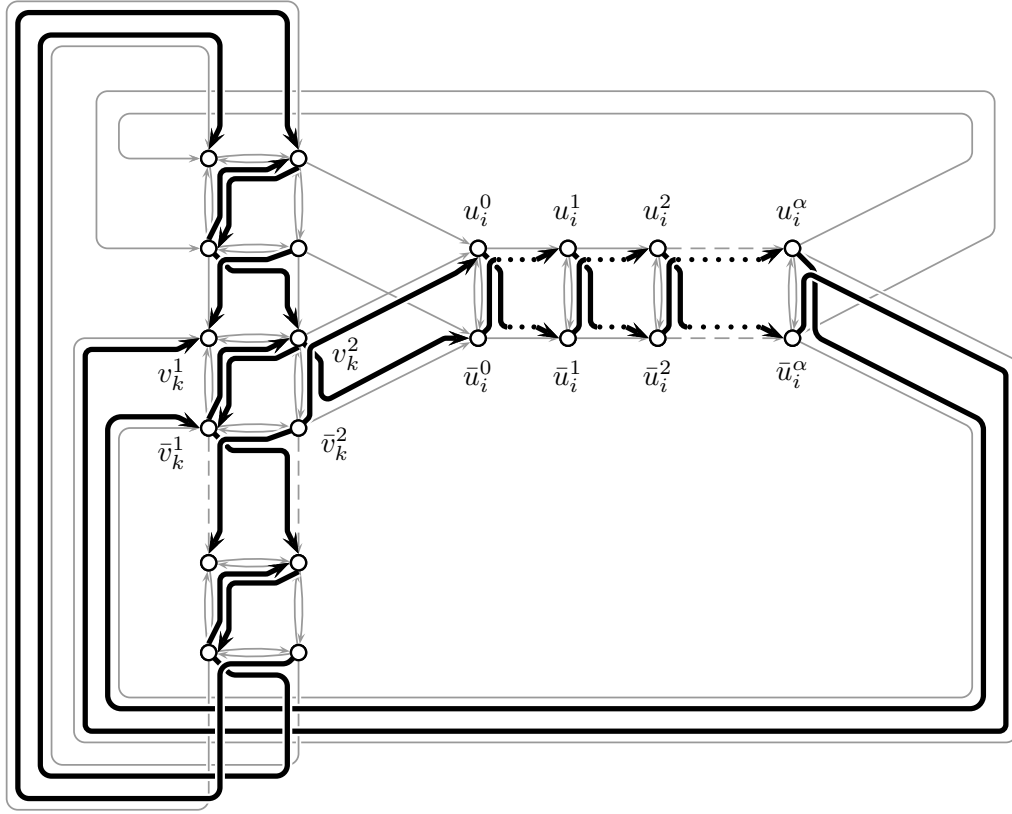
Figure 5.18 illustrates this forwarding.

Clearly, this construction is polynomial in the size of the given graph  $H$ .

For any path set  $R \subseteq S$ , we denote by  $E(R) := F \cap \mathcal{F}(R)$  the subset of the constructed forwarding that corresponds to  $R$ . In the reverse direction, we denote for each  $E \subseteq F$  by  $R(E) \subseteq S$  the smallest path set  $R$  with  $E \subseteq \mathcal{F}(S)$ .

The following argumentation bases on a simple observation:

**Observation (5.32a):** *Let  $i \in I$ ,  $l \in \{0, \dots, \alpha - 1\}$ , and  $\lambda \in \mathbb{R}_+^A$ . Then path  $(u_i^l, \bar{u}_i^l, \bar{u}_i^{l+1})$  is the unique shortest  $(u_i^l, \bar{u}_i^{l+1})$ -path with respect to  $\lambda$  if and only if arc  $(u_i^l, \bar{u}_i^l)$  is contained in the (unique) shortest  $(s_P, t_P)$ -path*



**Figure 5.18** Constructed forwarding  $F$ : For each shown path  $P$ , all pairs  $(t_P, a)$  with solid drawn arcs  $a \in P$  belong to  $F$ . Those pairs  $(t_P, a)$  with dashed drawn arcs  $a \in P$  do not belong to  $F$ .

with respect to  $\lambda$ .

Observation 5.32a implies that a metric  $\lambda$  is compatible with a path set  $R \subseteq S$  if and only if it is compatible with the corresponding forwarding  $E(R)$ . Thus, most parts of the proof of Theorem 5.15 carry over immediately.

First, suppose  $C \subseteq W$  is a vertex cover in  $H$ . Then the path set  $R(C)$  constructed in the proof of Theorem 5.15 is a non-USPS. By Observation 5.32a, the corresponding forwarding  $E(R(C))$  then is a (not necessarily irreducible) non-USPF. The size of  $E(R(C))$  clearly is

$$|E(R(C))| = 2\alpha |C| + 16m .$$

For the minimum vertex cover  $C^*$  and the minimum cardinality non-USPF  $E^* \subseteq F$  we therefore have

$$|E^*| \leq 2\alpha |C^*| + 16m . \quad (5.16)$$

For the other direction of the proof, suppose  $E \subseteq F$  is an irreducible non-USPF. Then Observation 5.32a implies that  $R(E)$  is a non-USPS. Furthermore, it follows that for each  $i \in I$  either  $F_i^1 \subset E$  or  $F_i^1 \cap E = \emptyset$ :

If  $E$  contained some but not all  $(t, a)$  pairs of some set  $F_i^1$ , then also the corresponding path set  $R(E)$  would contain some but not all paths of  $S_i^1$ . In the proof of Theorem 5.15, however, it was shown that if this were the case then either  $R(E)$  is a USPS or  $R(E)$  is reducible. This, in turn, would imply that also  $E$  is either a USPF or reducible, which contradicts our assumption.

Hence, for each irreducible non-USPF  $E \subseteq F$  we can properly define a corresponding vertex cover as

$$C(E) := \{w_i : F_i^1 \subset E\} .$$

One easily observes that this vertex cover  $C(E)$  is exactly the vertex cover  $C(R(E))$  defined for the corresponding path set  $R(E)$  in the proof of Theorem 5.15. From the above definition of  $C(E)$  we get that

$$|E| \geq 2\alpha |C(E)| . \tag{5.17}$$

With  $\alpha \rightarrow \infty$ , it now follows analogous to the proof of Theorem 5.15 from (5.16) and (5.17) that approximating MIN-NON-USPF is at least as hard as approximating MINIMUM VERTEX COVER.  $\square$

Theorem 5.32 carries over immediately to the weighted problem version.

**Corollary 5.33** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate MIN-WEIGHT-NON-USPF within a factor of  $7/6 - \epsilon$ .*

In Chapter 7 of this thesis we discuss an integer linear programming formulation for unsplittable shortest path routing problems that is based on binary arc routing variables for each commodity and the independence system characterization of unique shortest path forwardings. Corollary 5.33 will be important in this context, because the separation problem for one type of the original model inequalities in these formulations reduces to MIN-WEIGHT-NON-USPF.





## Part II

# Hardness and Approximability



## Chapter 6

# On the Approximability of Unsplittable Shortest Path Routing Problems

Many data networks presently employ shortest path routing protocols such as OSPF or IS-IS [54, 143]. With these routing protocols, all end-to-end traffic streams are routed along shortest paths with respect to some administrative link lengths. The simplicity of this policy offers many advantages in practice: It admits the use of decentralized and distributed routing algorithms, it has very good scaling properties with respect to the network size, and it typically leads to less administrative overhead than connection oriented routing schemes. On the other hand, the shortest path routing policy has an inherent drawback: It is not possible to configure the end-to-end routing paths between different terminal pairs individually. The routing paths can be controlled only jointly and only indirectly by changing the administrative routing lengths of the network links.

Finding routing lengths that induce a set of globally efficient end-to-end routing paths is a major difficulty in such networks. The shortest path routing paradigm enforces rather complicated and subtle interdependencies among the paths that comprise a valid routing. Additional difficulties arise if each communication demand must be sent unsplit through the network – a requirement that is often imposed in practice to ensure traceability of end-to-end traffic flows and to prevent package reordering and other unwanted effects of multi-path routing. In this case, the lengths must be chosen such that the shortest paths are uniquely determined for all communication demands. In practice, these routing restrictions may lead to unbalanced traffic flows with some highly congested network links. As this has severe effects on the overall service quality, network providers put much effort into reducing and balancing the congestion and into (re-)designing their networks with respect to this routing policy.

In this chapter, we discuss the relation between unsplittable shortest path

routing (USPR) and other routing schemes and we study the approximability of the three USPR network planning problems MIN-CON-USPR, CAP-USPR, and FC-USPR introduced in Chapter 3.

The congestion minimization problem MIN-CON-USPR corresponds to the task of finding an efficient USPR in an existing network. In this problem, we are given a digraph  $D = (V, A)$  with fixed arc capacities and a set  $K \subseteq V \times V$  of directed commodities with associated demand values, and we seek for a USPR that minimizes the maximum congestion (i.e., the flow to capacity ratio) over all arcs. The maximum congestion is a good measure for the overall network service quality.

The fixed charge network design problem FC-USPR and the capacitated network design problem CAP-USPR model the tasks of designing the topology and of dimensioning the link capacities of a USPR network, respectively. In CAP-USPR we wish to install integer multiples of some basic arc capacity units such that the resulting capacities admit a USPR of the given commodities and the total capacity installation cost is minimized, while in FC-USPR the goal is to find a minimum cost subgraph of  $D$  such that the given arc capacities admit a USPR in this subgraph.

All three problems are of great practical interest in the planning of networks that employ shortest path routing protocols.

This chapter is organized as follows. In Section 6.1, we briefly review some previous work concerning shortest path routing in general and the approximability of shortest path routing problems in particular. A (review of the) formal definition of the three USPR problems MIN-CON-USPR, CAP-USPR, and FC-USPR and a discussion of their basic properties follows in Section 6.2.

Section 6.3 contains a comparison of USPR with several other routing schemes. We construct examples where the minimum congestion that can be obtained with USPR is 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, and a factor of  $\Omega(|V|)$  larger than the congestion of an optimal unsplittable source-invariant routing. Furthermore, we show that the so-called no-bottleneck condition, which is typically assumed in unsplittable flow problems, has no effect on the complexity of unsplittable shortest path routing problems. This gives theoretical evidence for the practical experience that routing planning is harder for USPR than for the other routing schemes.

New strong hardness results for the three USPR problems are presented in Section 6.4. We prove that it is  $\mathcal{NP}$ -hard to approximate the minimum congestion problem MIN-CON-USPR within a factor of  $\mathcal{O}(|V|^{1-\epsilon})$  for any  $\epsilon > 0$  and that the fixed charge network design problem FC-USPR is  $\mathcal{NPO}$ -complete. Furthermore, we show that the capacitated network design problem CAP-USPR is  $\mathcal{NP}$ -hard to approximate within a factor of  $\mathcal{O}(2^{\log^{1-\epsilon}|V|})$  in the directed case or within a factor of  $2 - \epsilon$  in the undirected

Problem	Hardness	Approximability
MIN-CON-USPR	$\Omega( V ^{1-\epsilon})$	general: $\min\{ A ,  K \}$ undir. cycle: 2 bidir. ring: 3
CAP-USPR	undir.: $2 - \epsilon$ directed: $\Omega(1^{\log^{1-\epsilon}  V })$	general: — undir. cycle: 2 bidir. ring: 4 uniform: $ K $ single-source: $ K $ undir. uniform: $\mathcal{O}(\log  V )$
FC-USPR	$\mathcal{NPO}$ -complete, i.e., $\Omega(2^{ V ^{1-\epsilon}})$	—

Table 6.1: Approximability of USPR problems.

case.

In Section 6.5, we discuss polynomial time approximation algorithms for MIN-CON-USPR and CAP-USPR that are applicable for general underlying graphs and digraphs. In the first part of this section we devise simple  $|A|$ - and  $|K|$ -approximation algorithms for MIN-CON-USPR. In the second part we show how to approximate the uniform and the single-source CAP-USPR problem within a factor  $\mathcal{O}(|K|)$  and the undirected uniform CAP-USPR problem within a factor of  $\mathcal{O}(\log |V|)$ , using techniques that have been proposed in the literature for other capacitated network design problems.

In Section 6.6, we finally present constant factor approximation algorithms for MIN-CON-USPR and CAP-USPR for the special cases where the underlying graph is a bidirected ring or an undirected cycle. FC-USPR remains  $\mathcal{NPO}$ -complete even in these special cases.

Table 6.1 summarizes the results of this chapter. Unless stated otherwise, the hardness results also hold and the approximation algorithms also apply for the undirected problem versions, where both the underlying graph and the commodities are undirected.

## 6.1 Related Work

In spite of its long history in practice, unsplittable shortest path routing has received attention in the mathematical literature only recently.

Ben-Ameur and Gourdin [26, 27] and Broström and Holmberg [45, 46] studied structural properties of (undirected) path sets where all paths are uniquely determined shortest paths for edge metric. Ben-Ameur and Gourdin also devised integer linear programming models to find a metric that induces a prescribed set of shortest paths (or prove that no such metric ex-

ists). Farago et al. [85, 86] studied a special case of this INVERSE SHORTEST PATHS problem where the given paths are known to be shortest paths with respect to the number of edges and the task is to find lengths such that all these paths are unique shortest paths. Bley [35] proved that finding such a metric is computationally hard if the range of admissible link lengths is bounded.

Algorithms based on local search techniques, Lagrangian relaxation, and integer programming methods as well as computational results for real-world network design and congestion minimization problems with USPR and multi-shortest path routing have been discussed in [28, 34, 37, 36, 48, 81, 85, 92, 93, 115, 133, 163, 169]. Results concerning the approximability of USPR problems have not been published so far (to our knowledge). However, several related problems have already been studied.

Fortz and Thorup [93] proved that it is  $\mathcal{NP}$ -hard to approximate the minimum congestion that can be obtained with a multi-shortest path routing that splits traffic according to the ECMP rule (see Chapter 3) within a factor less than  $3/2$ . Dinitz et al. [74], Kolliopoulos and Stein [124], and Skutella [179] studied the approximability of (variants of) the unsplittable flow problem, while Lorenz et al. [134] discussed the relation of source-invariant routing to several other routing schemes.

Dodis and Khanna [75] proved that the directed GENERALIZED STEINER NETWORK problem, which is closely related to the fixed charge network design problem FC-USPR, is not approximable within a factor of  $\mathcal{O}(2^{\log^{1-\epsilon}|V|})$ , unless  $\mathcal{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$ .

Based on the result of Bartal [19], Fakcharoenphol et al. [84], and Charikar et al. [57], Awerbuch and Azar [11] proposed an  $\mathcal{O}(\log|V|)$ -approximation algorithm for the undirected uniform BUY-AT-BULK NETWORK DESIGN problem. This algorithm can be applied directly to the undirected CAP-USPR problem with uniform capacities, where it achieves the same approximation guarantee. Similar, the approximation algorithms proposed by Guha et al. [108], Gupta et al. [110], Talwar [182], and Meyerson et al. [140] for the single-source Buy-at-bulk network design problem carry over to the corresponding CAP-USPR problem version.

Cosares and Saniee [63] and Schrijver et al. [177] proposed approximation algorithms for the undirected RING LOADING problem, which is equivalent to the minimum congestion unsplittable flow problem on a cycle with unit capacities. These algorithms are based on rounding the optimal solution of a linear relaxation of the problem. With an appropriate tie-breaking rule in the rounding stage and some extension to handle non-uniform capacities and the directed case, the same approach can be applied also to the unsplittable shortest path routing problems and leads to constant factor approximation algorithms for MIN-CON-USPR and CAP-USPR on undirected and bidirected rings.

## 6.2 Unsplittable Shortest Path Routing Problems

Let  $D = (V, A)$  be a directed graph with arc capacities  $u_a \in \mathbb{Z}_+$ ,  $a \in A$ , and let  $K \subseteq V \times V$  be a set of directed commodities with demand values  $d_{s,t} \in \mathbb{Z}_+$ ,  $(s, t) \in K$ . For each commodity  $(s, t) \in K$ , let  $\mathcal{P}(s, t)$  denote the set of all  $(s, t)$ -paths in  $D$ . Furthermore, let  $\mathcal{P}(K) := \bigcup_{(s,t) \in K} \mathcal{P}(s, t)$ . For any path  $P$ , we write  $a \in P$  or  $v \in P$  to indicate that the arc  $a \in A$  or the node  $v \in V$  occurs in  $P$ . The concatenation of two paths  $P_1 = (v_0^1, a_1^1, \dots, v_l^1)$  and  $P_2 = (v_0^2, a_1^2, \dots, v_k^2)$  with  $v_l^1 = v_0^2$  is denoted by  $P_1 \oplus P_2 := (v_0^1, a_1^1, \dots, v_l^1 = v_0^2, a_1^2, \dots, v_k^2)$ . For simplicity, we refer to a path  $P = (v_0, a_1, v_1, \dots, a_l, v_l)$  with only its node sequence  $P = (v_0, v_1, \dots, v_l)$  if the underlying digraph  $D$  is simple. For any path  $P$  and any arc length vector  $\lambda = (\lambda_a)_{a \in A} \in \mathbb{R}_+^A$ , we denote  $\lambda(P) := \sum_{a \in P} \lambda_a$ .

Recall that a metric  $\lambda \in \mathbb{R}_+^A$  is said to be *valid* for the commodity set  $K$  if, for each commodity  $(s, t) \in K$ , the shortest  $(s, t)$ -path  $P_{(s,t)}^*(\lambda)$  with respect to  $\lambda$  is uniquely determined. Given such a valid metric, the set of these induced shortest paths forms an *unsplittable shortest path routing (USPR)* for the commodity set  $K$ , see Definition 3.4 on Page 33. The demand of each commodity  $(s, t) \in K$  is routed unsplit along the respective shortest path  $P_{(s,t)}^*(\lambda)$  in a USPR. Given a valid metric  $\lambda$ , the total flow through an arc  $a \in A$  in the induced USPR is

$$f_a(\lambda) := \sum_{(s,t) \in K: a \in P_{(s,t)}^*(\lambda)} d_{s,t}. \quad (6.1)$$

The task in the minimum congestion unsplittable shortest path routing problem MIN-CON-USPR is to find a valid metric  $\lambda \in \mathbb{Z}_+^A$  for the given commodity set  $K$ , such that the induced flows minimizes the maximum of the congestion values  $f_a(\lambda)/u_a$  over all arcs. Formally, this problem has been defined as follows:

<i>Problem:</i>	MIN-CON-USPR
<i>Instance:</i>	A digraph $D = (V, A)$ with arc capacities $u_a \in \mathbb{Z}_+$ , $a \in A$ , and a commodity set $K \subseteq V \times V$ with demands $d_{s,t} \in \mathbb{Z}_+$ , $(s, t) \in K$ .
<i>Solution:</i>	A valid metric $\lambda \in \mathbb{Z}_+^A$ for the commodity set $K$ , i.e., such that the shortest $(s, t)$ -path w.r.t. $\lambda$ is uniquely determined for each commodity $(s, t) \in K$ .
<i>Objective:</i>	$\min\{L \in \mathbb{R}_+ : f_a(\lambda) \leq L u_a \text{ for all } a \in A\}$ .

In the network design and dimensioning problems, we are not only given arc capacities  $u_a \in \mathbb{Z}_+$ ,  $a \in A$ , but also arc costs  $w_a \in \mathbb{Z}_+$ ,  $a \in A$ .

In the capacitated network design problem CAP-USPR, we seek for integer capacity multipliers  $z_a \in \mathbb{Z}_+$ ,  $a \in A$ , and for a valid metric  $\lambda \in \mathbb{Z}_+^A$  for

the commodity set  $K$ , such that  $f_a(\lambda) \leq u_a z_a$  for all  $a \in A$ . The objective is to minimize the total capacity installation cost  $\sum_{a \in A} w_a z_a$ .

<i>Problem:</i>	CAP-USPR
<i>Instance:</i>	A digraph $D = (V, A)$ with arc capacities $u_a \in \mathbb{Z}_+$ , $a \in A$ , and arc costs $w_a \in \mathbb{Z}_+$ , $a \in A$ , and a commodity set $K \subseteq V \times V$ with demands $d_{s,t} \in \mathbb{Z}_+$ , $(s, t) \in K$ .
<i>Solution:</i>	Capacity multipliers $z \in \mathbb{Z}_+^A$ and a valid metric $\lambda \in \mathbb{Z}_+^A$ for the commodity set $K$ such that $f_a(\lambda) \leq z_a u_a$ for all $a \in A$ .
<i>Objective:</i>	$\min \sum_{a \in A} z_a w_a$ .

In the fixed charge network design problem FC-USPR, the given capacities are regarded as fix. The task is to find a minimum cost arc set  $B \subseteq A$  and a metric  $\lambda \in \mathbb{Z}_+^B$ , such that  $\lambda$  defines a USPR for the commodities  $K$  in the subgraph  $(V, B)$  and the induced arc flows  $f_a(\lambda)$  do not exceed the capacities  $u_a$  (on the arcs  $a \in B$ ).

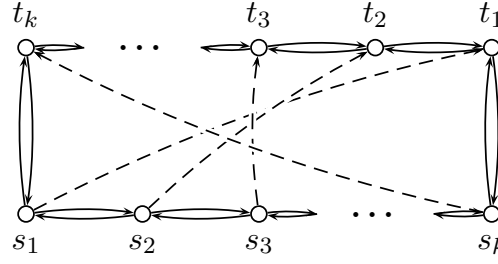
<i>Problem:</i>	FC-USPR
<i>Instance:</i>	A digraph $D = (V, A)$ with arc capacities $u_a \in \mathbb{Z}_+$ , $a \in A$ , and arc costs $w_a \in \mathbb{Z}_+$ , $a \in A$ , and a commodity set $K \subseteq V \times V$ with demands $d_{s,t} \in \mathbb{Z}_+$ , $(s, t) \in K$ .
<i>Solution:</i>	An arc set $B \subseteq A$ and a metric $\lambda \in \mathbb{Z}_+^B$ that is valid in the subgraph $(V, B)$ for the commodity set $K$ , such that $f_a(\lambda) \leq u_a$ for all $a \in B$ .
<i>Objective:</i>	$\min \sum_{a \in B} w_a$ .

In all three problems, we may assume without loss of generality that  $D$  contains an  $(s, t)$ -path for each commodity  $(s, t) \in K$ , that  $D$  contains no loops, that  $u_a > 0$  for all  $a \in A$ , and that  $d_{s,t} > 0$  for all  $(s, t) \in K$ . Also, we assume that there are no parallel commodities: If there were two or more parallel commodities from  $s$  to  $t$ , these would have to use the same (uniquely determined shortest) flow-path in any unsplittable shortest path routing and, therefore, could be aggregated into one commodity.

For notational simplicity, we also assume in the following that the routing metric  $\lambda$  in a solution of the fixed charge problem FC-USPR is defined not only on the arc set  $B$  describing the solution's topology but on the entire arc set  $A$ . Clearly, any metric  $\lambda \in \mathbb{Z}_+^B$  that induces a unique shortest  $(s, t)$ -path in  $(V, B)$  for all  $(s, t) \in K$  can be extended to a metric  $\lambda \in \mathbb{Z}_+^A$  that induces exactly the same  $(s, t)$ -path for all  $(s, t) \in K$  in  $D = (V, A)$  by setting  $\lambda_a = \sum_{b \in B} \lambda_b + 1$  for all  $a \in A \setminus B$ .

One easily observes that, for any bijection  $\text{idx} : A \leftrightarrow \{1, \dots, |A|\}$ , the metric  $\lambda_a := 2^{\text{idx}(a)}$  induces unique shortest paths between all connected





**Figure 6.1** Reduction from PARTITION to MIN-CON-USPR: Solid lines are arcs, dashed lines are commodities.

node pairs. Hence, any instance of MIN-CON-USPR or of CAP-USPR has a feasible solution, provided that the underlying digraph  $D$  contains at least one  $(s, t)$ -path for each  $(s, t) \in K$ .

If the underlying graph  $D$  contains *only* one  $(s, t)$ -path for each  $(s, t) \in K$ , then all metrics  $\lambda \in \mathbb{Z}_+^A$  define the same USPR. In this case, any metric defines an optimal solution for MIN-CON-USPR and CAP-USPR, and FC-USPR is trivially solvable. The simplest non-trivial case is when the underlying digraph contains two paths for each commodity. Already in this case all three USPR problems become (weakly)  $\mathcal{NP}$ -hard.

**Theorem 6.1** MIN-CON-USPR, FC-USPR, and CAP-USPR are  $\mathcal{NP}$ -hard, even if the underlying digraph is a bidirected ring.

**Proof.** We present a polynomial reduction from the PARTITION problem to the problem of solving MIN-CON-USPR to optimality. The  $\mathcal{NP}$ -hardness of FC-USPR and CAP-USPR follows analogously. A similar construction also can be used to show the (weak)  $\mathcal{NP}$ -hardness of the more general unsplittable flow and ring loading problems, see Skutella [179] and Cosares and Saniee [63] for example. Given a set of items  $i \in \{1, \dots, k\}$  with sizes  $d_i \in \mathbb{Z}_+$ , the PARTITION problem is to find a subset  $S \subseteq \{1, \dots, k\}$  with  $\sum_{i \in S} d_i = 1/2 \sum_{i=1}^k d_i$  or to prove that no such subset exists. This problem is known to be  $\mathcal{NP}$ -complete; see Karp [121] or Garey and Johnson [96].

Given a PARTITION instance consisting of the items  $i \in \{1, \dots, k\}$  with sizes  $d_i \in \mathbb{Z}_+$ , the instance of MIN-CON-USPR is built as shown in Figure 6.1. For each item  $i \in \{1, \dots, k\}$ , we introduce two nodes  $s_i, t_i$  and a commodity  $(s_i, t_i)$  with a demand value  $d_{s_i, t_i} = d_i$ . The arc set consists of the arcs  $(s_i, s_{i+1})$ ,  $(s_{i+1}, s_i)$ ,  $(t_i, t_{i+1})$ , and  $(t_{i+1}, t_i)$  for all  $i = 1, \dots, k-1$ , as well as  $(s_1, t_k)$ ,  $(t_k, s_1)$ ,  $(s_k, t_1)$ , and  $(t_1, s_k)$ . The arc capacities are set to

$$u_a := \begin{cases} 1/2 \sum_{i=1}^k d_i, & \text{if } a \in \{(s_1, t_k), (t_k, s_1), (s_k, t_1), (t_1, s_k)\}, \\ \sum_{i=1}^k d_i, & \text{otherwise.} \end{cases}$$

It is not difficult to verify that any unsplittable flow routing of these commodities is also an unsplittable shortest path routing. Therefore, any feasible partition of the items corresponds to an unsplittable shortest path routing

of the commodities such that the flows do not exceed the arc capacities, and vice versa. The commodities routed across arc  $(s_1, t_k)$  form one set of the partition, those routed across  $(s_k, t_1)$  the other set.  $\square$

One easily observes that all three USPR problems also contain the DISJOINT PATHS problem as a special case. For general directed graphs, these problems therefore are actually  $\mathcal{NP}$ -hard in the strong sense, even if all demands and capacities are equal to one.

### 6.3 Relation to Other Routing Schemes and the No-Bottleneck Condition

The unsplittable shortest path routing model is very restrictive and inherits structural properties of several other routing models. In this section, we compare USPR to four closely related but less restrictive routing models. We show that the minimal congestion that can be obtained with USPR for a given commodity set may exceed the congestion achievable with the other routing models by an arbitrarily large factor.

The most flexible routing model is (fractional) multicommodity flow routing. With this routing model, the demand of each commodity may be distributed arbitrarily and independent of the other commodities onto several flow paths. It thus admits the best possible use of the available capacities. In order to implement MCF routing in practice, the network must admit the configuration of arbitrary end-to-end routing paths and flow distributions for each commodity individually. This introduces many practical difficulties and complicates the network management. Therefore, many telecommunication network protocols are based on routing models that are less capacity efficient but easier to implement in practice.

With shortest multi-path routing, the traffic that is sent from a node  $s$  to a node  $t$  is distributed equally to all neighbors of  $s$  that are contained in any shortest  $(s, t)$ -path with respect to some metric  $\lambda$ . This routing model adequately describes so-called equal cost multi-path traffic splitting policies in shortest path routing protocols. Fortz and Thorup [93] show that the minimum congestion achievable with shortest multi-path routing cannot be approximated within a factor less than  $3/2$ .

The unsplittable flow routing model requires that each commodity is sent unsplit via a single path through the network. In contrast to the USPR model, the commodities' flow paths may be chosen independent of each other. Kolliopoulos and Stein [124] prove that it is  $\mathcal{NP}$ -hard to approximate the minimum congestion unsplittable flow routing within  $2 - \epsilon$ , for any  $\epsilon > 0$ .

The fourth routing model that is closely related to USPR is unsplittable source-invariant routing. With this routing model, each commodity is routed on a single flow path. All flow paths with the same destination must form an

anti-arborescence directed towards this destination. Once two flows meet on their way to a common destination, they cannot split anymore. This model describes the routing possibilities of packet networks with independently configurable store-and-forward routers. Lorenz et. al. [134] show that finding a minimum congestion unsplittable source-invariant routing is  $\mathcal{NP}$ -hard. They also show that the minimum congestion may be factor  $\Omega(|V|)$  higher for unsplittable source-invariant routing than for unsplittable flow routing.

Given a digraph  $D = (V, A)$  with arc capacities  $u_a \in \mathbb{Z}_+$ ,  $a \in A$ , and commodities  $K \subseteq V \times V$  with demand values  $d_{s,t} \in \mathbb{Z}_+$ ,  $(s, t) \in K$ , we denote the optimal solution value of the MIN-CON-USPR problem by  $L^{USPR}$ . With  $L^{MCF}$ ,  $L^{UFP}$ ,  $L^{SMPR}$ , and  $L^{USIR}$  we refer to the minimal congestion values that can be obtained with fractional multicommodity flow routing (MCF), an unsplittable flow routing (UFP), a shortest multi-path routing (SMPR), and an unsplittable source-invariant routing (USIR) on the same instance, respectively. It is obvious that

$$\begin{aligned} L^{USPR} &\geq L^{UFP} \geq L^{MCF}, \\ L^{USPR} &\geq L^{SMPR} \geq L^{MCF}, \text{ and} \\ L^{USPR} &\geq L^{USIR} \geq L^{MCF}, \end{aligned}$$

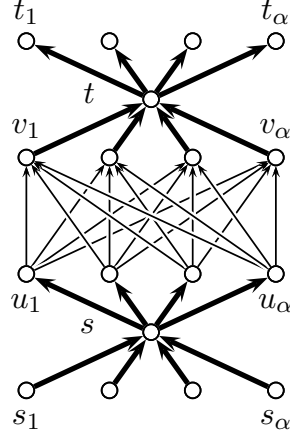
since every unsplittable shortest path routing is also a valid shortest multi-path routing, a valid unsplittable flow routing, and a valid unsplittable source-invariant routing of the given commodities. In the following we construct instances where the gap between USPR and the other routing models becomes arbitrarily large.

**Proposition 6.2** *There is a family of instances with*

- (i)  $L^{USPR} \geq \Omega(|V|^2) \cdot L^{SMPR}$ ,
- (ii)  $L^{USPR} \geq \Omega(|V|^2) \cdot L^{UFP}$ , and
- (iii)  $L^{USPR} \geq \Omega(|V|^2) \cdot L^{MCF}$ .

**Proof.** Let  $\alpha \in \mathbb{N}$  and consider the digraph  $D = (V, A)$  illustrated in Figure 6.2. It consists of the nodes  $V := \{s, t\} \cup \{s_i, t_i, u_i, v_i : i = 1, \dots, \alpha\}$  and the arcs  $A := A_1 \cup A_2$ , where  $A_1 := \{(u_i, v_j) : i, j = 1, \dots, \alpha\}$  and  $A_2 := \{(s_i, s), (s, u_i), (v_i, t), (t, t_i) : i = 1, \dots, \alpha\}$ . The arc capacities are  $u_a = 1$  for all  $a \in A_1$  and  $u_a = \alpha$  for all  $a \in A_2$ . In this graph, consider the commodities  $K := \{(s_i, t_j) : i, j = 1, \dots, \alpha\}$  with demands  $d_{s_i, t_j} = 1$  for all  $(s_i, t_j) \in K$ .

The congestion of any unsplittable shortest path routing is  $L^{USPR} = \alpha^2$ , since all commodities' routing paths must follow the same subpath between the nodes  $s$  and  $t$  in an unsplittable shortest path routing, and therefore share some arc  $(u_i, v_j)$  of capacity 1.



**Figure 6.2** Instance with  $L^{USPR} = \alpha^2$  and  $L^{UFP} = L^{SMPR} = L^{MCF} = 1$ .

Arcs with capacity  $\alpha$  are bold, arcs with capacity 1 are thin. For each node pair  $(s_i, t_j)$  there is a commodity with demand 1.

On the other hand, the congestion is 1 for an optimal shortest multi-path routing (where all arc lengths are chosen equal), as well as for an optimal unsplittable flow routing or an optimal multicommodity flow routing. For  $\alpha \rightarrow \infty$ , we obtain the claimed relations.  $\square$

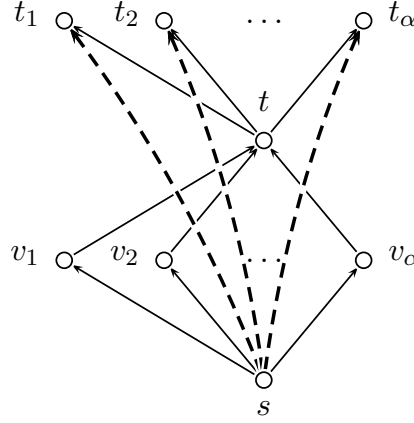
**Proposition 6.3** *There is a family of instances with*

$$L^{USPR} \geq \Omega(|V|) \cdot L^{USIR}.$$

**Proof.** Let  $\alpha \in \mathbb{Z}_+$  and consider the digraph illustrated in Figure 6.3. It contains the nodes  $V := \{s, t\} \cup \{t_i, v_i : i = 1, \dots, \alpha\}$  and the arcs  $A := \{(s, v_i), (v_i, t), (t, t_i) : i = 1, \dots, \alpha\}$ . All arcs have capacity 1. In this network, consider the commodities  $K := \{(s, t_i) : i = 1, \dots, \alpha\}$  with demands  $d_{s, t_i} = 1$ . In any unsplittable shortest path routing, all commodities are routed via the same subpath between  $s$  and  $t$ . The minimal congestion value for USPR therefore is  $L^{USPR} = \alpha$ . With source-invariant routing, the commodities may be routed via different  $s, t$ -subpaths, as they have different destinations. The optimal congestion for this routing model therefore is 1, and with  $\alpha \rightarrow \infty$  we obtain the claim.  $\square$

The presented worst-case gaps between the different routing paradigms hold for the corresponding undirected routing variants, too. (In the undirected shortest multi-path routing policy and the undirected unsplittable source invariant routing policy, we arbitrarily choose  $s$  as the source and  $t$  as the destination for each undirected commodity  $(s, t) \in K$ .)

An assumption commonly made for unsplittable flow problems is that the maximum demand value does not exceed the minimum capacity. Typically,



**Figure 6.3** Instance with  $L^{USPR} \geq \Omega(|V|) L^{USIR}$ . The solid lines are arcs, the dashed lines are the commodities. All capacities and demand values are one.

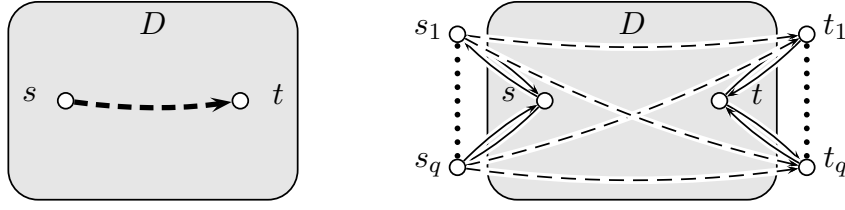
unsplittable flow problems are easier to approximate if this additional condition holds than in the general case; see Dinitz et al. [74], Kolliopoulos and Stein [124], or Skutella [179], for example. For unsplittable shortest path routing problems, however, this so-called no-bottleneck condition has no effect on the approximability.

**Proposition 6.4** *For any instance  $I = (D, u, K, d)$  of MIN-CON-USPR with  $d_{\max} > u_{\min}$ , there exists an equivalent instance  $I' = (D', u', K', d')$  with  $d'_{\max} \leq u'_{\min}$  (i.e., any solution for  $I$  with objective value  $L$  can be transformed into a solution for  $I'$  with objective value  $L$ , and vice versa).*

**Proof.** Suppose we are given a MIN-CON-USPR instance  $(D, u, K, d)$  with  $d_{\max} > u_{\min}$ . Let  $r := d_{\max}/u_{\min}$  and  $q := \lceil \sqrt{r} \rceil$ . For each node  $v \in V$ , we introduce  $q$  additional nodes  $v_j$  and  $2q$  arcs  $(v, v_j)$  and  $(v_j, v)$ ,  $j = 1, \dots, q$ , see Figure 6.4. The capacities  $u'$  are given as  $u'_a := u_a$  for all  $a \in A$ , and  $u'_{(v, v_j)} = u'_{(v_j, v)} := \sum_{a \in A} u_a$  for all  $j = 1, \dots, q$ . Each commodity  $(s, t) \in K$  is replaced by  $q^2$  new commodities  $(s_i, t_j)$  with demand values  $d'_{s_i, t_j} \simeq d_{s, t}/q^2$ . (Or more precisely, with demand values  $d'_{s_i, t_j} \in \{\lfloor d_{s, t}/q^2 \rfloor, \lceil d_{s, t}/q^2 \rceil\}$  such that  $\sum_{i, j=1}^q d'_{s_i, t_j} = d_{s, t}$ .)

Clearly,  $d'_{\max} \leq u'_{\min}$  holds.

Now, consider the set of all commodities  $(s_i, t_j)$ ,  $i, j = 1, \dots, q$ , for some  $(s, t) \in K$ . Since all nodes  $s_i$  have only one neighbor  $s$  and all nodes  $t_j$  have only one neighbor  $t$ , all these  $q^2$  commodities  $(s_i, t_j)$  must be routed via the same  $(s, t)$ -subpath in an unsplittable shortest path routing. Therefore, any unsplittable shortest path routing of the commodities  $K$  in  $D$  corresponds to an unsplittable shortest path routing of the commodities  $K'$  in  $D'$ , and vice versa. As the corresponding routings induce the same flows on the arcs of  $D$ , the maximum congestion values are equal for both routings.  $\square$



**Figure 6.4** The no-bottleneck condition is irrelevant for USPR problems: The large commodity  $(s, t)$  can be replaced by many small commodities  $(s_i, t_j)$ , which must share the same  $(s, t)$ -path.

Analogously, any instance of FC-USPR or of CAP-USPR can be transformed into an equivalent instance with  $d_{\max} \leq u_{\min}$ . However, note that in general this transformation is not polynomial in the strong sense, because the size of the underlying digraph grows by a factor of  $\Theta(d_{\max}/u_{\min})$ .

## 6.4 Inapproximability Results

In the following section, we analyze how hard it is to approximate the three unsplittable shortest path routing problems.

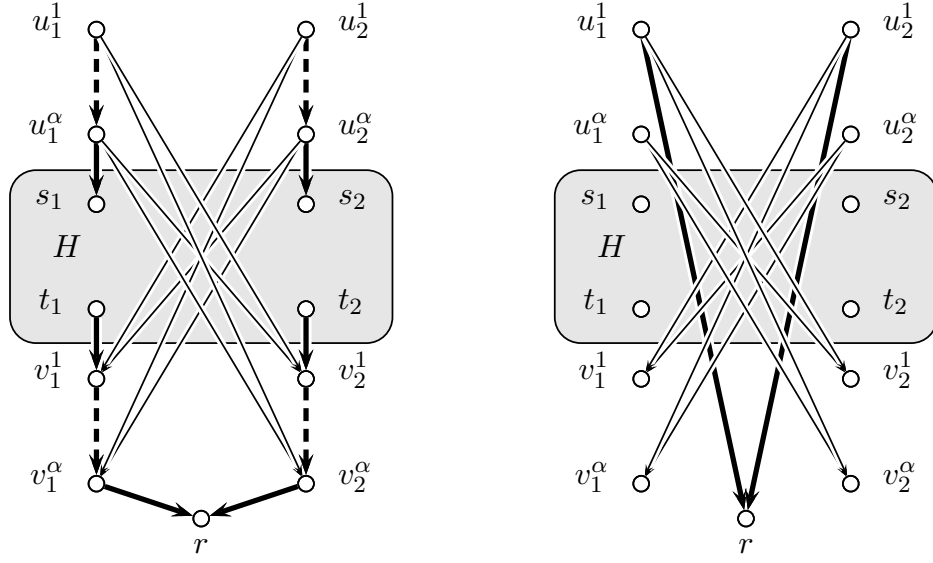
### 6.4.1 Congestion Minimization

We begin by showing that MIN-CON-USPR is not approximable within a factor of  $\mathcal{O}(|V|^{1-\epsilon})$ , unless  $\mathcal{P} = \mathcal{NP}$ . As a first step, we show that there is no constant factor approximation.

**Lemma 6.5** *Let  $\alpha \in \mathbb{Z}_+$  be an arbitrary number. It is  $\mathcal{NP}$ -hard to approximate MIN-CON-USPR within a factor less than  $\alpha + 1$ .*

**Proof.** We construct a reduction from the  $\mathcal{NP}$ -complete decision problem FULLY DISJOINT PATHS to MIN-CON-USPR. FULLY DISJOINT PATHS is a restricted variant of the classical DISJOINT PATHS problem [96]. Given a directed graph  $H = (W, F)$  and a set of node pairs  $(s_i, t_i)$ ,  $i = 1, \dots, k$ , the task is to find  $(s_i, t_i)$ -paths  $P_i$  in  $H$  that are not only internally disjoint but share no nodes at all (including the paths' first and last nodes), i.e.,  $\{v \in W : v \in P_i \text{ and } v \in P_j\} = \emptyset$  for all  $i, j = 1, \dots, k$  with  $i \neq j$ . It is easy to verify that the DISJOINT PATHS problem remains  $\mathcal{NP}$ -complete even with this stronger notion of disjointness. Note that the directed version of FULLY DISJOINT PATHS remains  $\mathcal{NP}$ -hard even if the number  $k \geq 2$  of nodes pairs is not part of the problem input. Yet, we assume that  $k$  is part of the input of FULLY DISJOINT PATHS, because then our construction carries over literally to the undirected problem version.

Suppose we are given a FULLY DISJOINT PATHS instance consisting of the digraph  $H = (W, F)$  and the node pairs  $(s_i, t_i)$ ,  $i = 1, \dots, k$ . W.l.o.g., we



(a) Digraph with indicated arc capacities: Arcs with capacity  $\alpha$  are bold, arcs with capacity 1 are thin.

(b) Commodities with demand values: commodities with demand  $\alpha$  are bold, commodities with demand 1 are thin.

**Figure 6.5** Constructed MIN-CON-USPR instance.

may assume that there is an  $(s_i, t_i)$ -path in  $H$  for each  $i = 1, \dots, k$  and that  $\{s_i, t_i\} \cap \{s_j, t_j\} = \emptyset$  for all  $i \neq j$ .

We construct a MIN-CON-USPR instance  $(D, u, K, d)$  as follows. The digraph  $D = (V, A)$  contains all nodes and arcs of  $H$ . Furthermore,  $D$  contains one extra node  $r$  and  $2k\alpha$  additional nodes  $u_i^l$  and  $v_i^l$  with  $l = 1, \dots, \alpha$  and  $i = 1, \dots, k$ , i.e.,

$$V := W \cup \{r\} \cup \{u_i^l, v_i^l : i = 1, \dots, k, l = 1, \dots, \alpha\}.$$

For each pair  $i, j = 1, \dots, k$  with  $i \neq j$  we add  $\alpha^2$  new arcs. These arcs form the set

$$A_1 := \{(u_i^l, v_j^m) : i, j = 1, \dots, k, i \neq j, l, m = 1, \dots, \alpha\}.$$

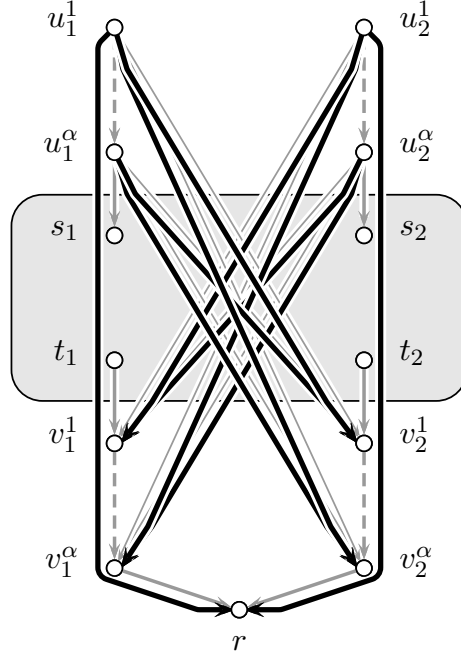
Additionally, we introduce  $2\alpha + 1$  arcs for each  $i = 1, \dots, k$ , which together comprise the arc set

$$A_2 := \{(u_i^\alpha, s_i), (t_i, v_i^1), (v_i^\alpha, r) : i = 1, \dots, k\} \\ \cup \{(u_i^l, u_i^{l+1}), (v_i^l, v_i^{l+1}) : i = 1, \dots, k, l = 1, \dots, \alpha - 1\}.$$

We let  $A := F \cup A_1 \cup A_2$ . The arc capacities are defined as

$$u_a := \begin{cases} 1, & \text{if } a \in A_1, \text{ and} \\ \alpha, & \text{otherwise.} \end{cases}$$

The commodity set  $K$  contains two types of commodities. For each  $i = 1, \dots, k$ , there is a commodity  $(u_i^1, r)$  with demand value  $d_{u_i^1, r} = \alpha$ . For



**Figure 6.6** USPR in  $D$  if fully disjoint  $(s_i, t_i)$ -paths exist in  $H$ .

each pair  $i, j = 1, \dots, k$  with  $i \neq j$  and each pair  $l, m = 1, \dots, \alpha$ , there is a commodity  $(u_i^l, v_j^m)$  with  $d_{u_i^l, v_j^m} = 1$ . Figure 6.5 illustrates the constructed MIN-CON-USPR instance for the case where  $k = 2$ .

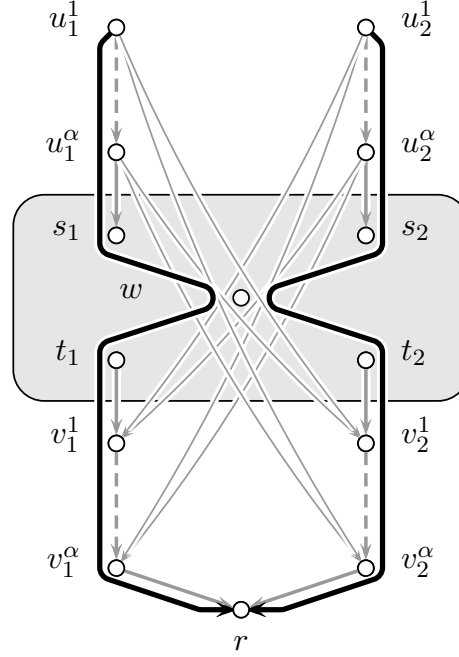
It is obvious that this transformation is polynomial in the encoding size of the given FULLY DISJOINT PATHS instance and  $\alpha$ . Furthermore, any metric  $\lambda \in \mathbb{Z}_+^A$  that induces unique shortest paths for the commodities in  $K$  defines a feasible solution for the constructed MIN-CON-USPR instance.

In the first part of the proof, we show that there exists an unsplittable shortest path routing whose induced flows do not exceed the arc capacities if the given FULLY DISJOINT PATHS instance has a feasible solution. Assume there exist fully disjoint  $(s_i, t_i)$ -paths  $P_i$  in  $H$ ,  $i = 1, \dots, k$ . Then we define the metric  $\lambda$  as

$$\lambda_a := \begin{cases} 1, & \text{if } a \in P_i \text{ for some } i \in \{1, \dots, k\} \text{ or} \\ & \text{if } a \in \bigcup_{i=1}^k \{(u_i^1, u_i^2), \dots, (u_i^\alpha, s_i), (t_i, v_i^\alpha), \dots, (v_i^2, v_i^1)\}, \\ |A|, & \text{otherwise.} \end{cases}$$

One easily finds that *all* shortest paths in  $D$  are unique with respect to  $\lambda$ . In particular, the shortest  $(u_i^1, r)$ -path is the path  $P_{(u_i^1, r)}^* = (u_i^1, \dots, u_i^\alpha, s_i) \oplus P_i \oplus (t_i, v_i^\alpha, \dots, v_i^1, r)$  for each  $i = 1, \dots, k$ , and the shortest  $(u_i^l, v_j^m)$ -path is  $P_{(u_i^l, v_j^m)}^* = (u_i^l, v_j^m)$  for each  $i, j = 1, \dots, k$  with  $i \neq j$  and  $l, m = 1, \dots, \alpha$ . Figure 6.6 illustrates this routing. It is not difficult to verify that the arc flows  $f_a(\lambda)$  induced by this routing do not exceed the arc capacities  $u_a$ ,  $a \in A$ .



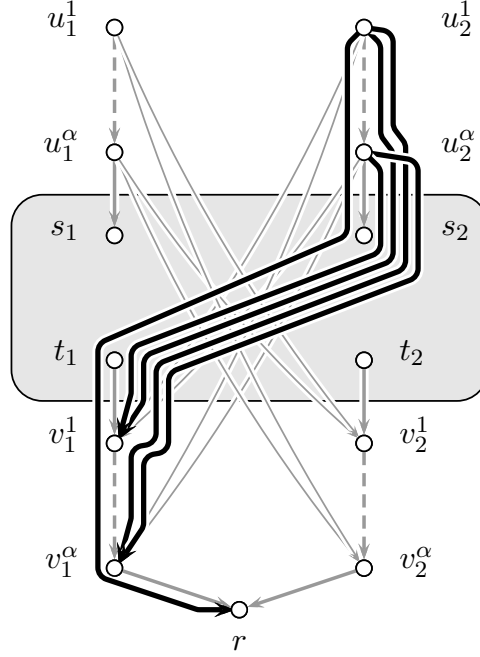


**Figure 6.7** No USPR. If the shortest  $(u_1^1, r)$ - and  $(u_2^1, r)$ -paths intersect in some internal node  $w$ , they are not uniquely determined.

In the second part of the proof, we show that the flows of any unsplittable shortest path routing exceed at least one arc capacity by a factor of at least  $\alpha + 1$  if the FULLY DISJOINT PATHS instance has no solution. So, suppose there is no set of fully disjoint  $(s_i, t_i)$ -paths in  $H$  and let  $\lambda \in \mathbb{Z}_+^A$  be an arbitrary metric that defines unique shortest  $(u, v)$ -paths  $P_{(u,v)}^*$  for all  $(u, v) \in K$ .

First, assume that some arc  $(u_i^l, v_j^m)$  is contained in the shortest path  $P_{(u_h^1, r)}^*$  for some commodity  $(u_h^1, r)$ . Since  $P_{(u_h^1, r)}^*$  is the unique shortest  $(u_h^1, r)$ -path, its arc  $(u_i^l, v_j^m)$  is also the unique shortest  $(u_i^l, v_j^m)$ -path. The total flow across this arc therefore is at least  $d_{u_h^1, r} + d_{u_i^l, v_j^m} = \alpha + 1$ , while its capacity is only  $u_{(u_i^l, v_j^m)} = 1$ . This leads to a congestion of at least  $\alpha + 1$  for this routing. In the following, we thus may assume that all commodities  $(u_i^1, r)$ ,  $i = 1, \dots, k$ , with demands  $d_{u_i^1, r} = \alpha$  are routed within the subgraph  $D' = (V, F \cup A_2)$ .

Now, suppose we had  $t_i \in P_{(u_i^1, r)}^*$  for all  $i = 1, \dots, k$ . Then at least two of these paths, say  $P_{(u_1^1, r)}^*$  and  $P_{(u_2^1, r)}^*$ , would have to intersect in some internal node  $w \in W$ , as illustrated in Figure 6.7. Otherwise, there would exist fully disjoint  $(s_i, t_i)$ -paths in  $H$ . However, these two paths  $P_{(u_1^1, r)}^*$  and  $P_{(u_2^1, r)}^*$  cannot be unique shortest paths w.r.t.  $\lambda$ , as they contain different subpaths between  $w$  and  $r$ .



**Figure 6.8** If  $t_1 \in P_{(u_2^1, r)}^*$ , then the shortest path property forces all commodities  $(u_2^l, v_1^m)$  to follow their respective subpath of  $P_{(u_2^1, r)}^*$ .

Consequently, there must be some  $i \in \{1, \dots, k\}$  such that  $t_i \notin P_{(u_i^1, r)}^*$ . Since  $P_{(u_i^1, r)}^*$  is completely contained in  $D'$ , there must be some  $j \neq i$  such that  $t_j \in P_{(u_i^1, r)}^*$ . Furthermore, all nodes  $u_i^l$  and  $v_j^m$ ,  $l, m = 1, \dots, \alpha$ , are contained in  $P_{(u_i^1, r)}^*$ . Hence, all  $\alpha^2$  commodities  $(u_i^l, v_j^m)$  are routed along their respective subpath of  $P_{(u_i^1, r)}^*$ , see Figure 6.8. The total flow across arc  $(t_j, u_j^\alpha)$  therefore is at least  $\alpha + \alpha^2$ , while its capacity is only  $\alpha$ .

Together, the two parts of the proof imply that it is  $\mathcal{NP}$ -hard to approximate MIN-CON-USPR within a factor less than  $\alpha + 1$ .  $\square$

If we choose  $\alpha = 1$  in the above construction, all capacities and demand values of the MIN-CON-USPR instance are equal to 1. This yields the following theorem.

**Theorem 6.6** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate MIN-CON-USPR within a factor of  $2 - \epsilon$ , even if all demand values and capacities are equal to one.*

For the general case, we obtain a stronger non-constant inapproximability bound by choosing  $\alpha$  depending on the size of the given FULLY DISJOINT PATHS instance.

**Theorem 6.7** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate MIN-CON-USPR within a factor of  $\mathcal{O}(|V|^{1-\epsilon})$ .*

**Proof.** The encoding size of the constructed MIN-CON-USPR instance is bounded by  $\mathcal{O}(\alpha^2 \log \alpha(|W| + |F|))$ . With  $\alpha = \alpha(H) := |W|^q$ , the presented construction thus remains polynomial in  $|W| + |F|$  for any fixed  $q \in \mathbb{N}$ . Because  $|V| \in \Omega(\alpha)$ , there exists some  $q_\epsilon \in \mathbb{Z}_+$  for every  $\epsilon > 0$ , such that  $\alpha \notin \mathcal{O}(|V|^{1-\epsilon})$  for  $\alpha := |W|^{q_\epsilon}$ . With Lemma 6.5, this implies the claim.  $\square$

Analogously, it follows that approximating MIN-CON-USPR within a factor of  $\mathcal{O}(|A|^{1/2-\epsilon})$  or  $\mathcal{O}(\langle I \rangle^{1/2-\epsilon})$  is  $\mathcal{NP}$ -hard for any  $\epsilon > 0$ , where  $\langle I \rangle$  is the encoding size of the MIN-CON-USPR instance (including the encoding size of the cost and capacity values).

By adding  $\alpha$  many new nodes  $r^j$ ,  $j = 1, \dots, \alpha$ , and replacing each commodity  $(u_i^1, r)$  of demand  $d_{u_i^1, r} = \alpha$  by  $\alpha$  many commodities  $(u_i^1, r^j)$  with  $d_{u_i^1, r^j} = 1$ , we may transform the MIN-CON-USPR instance constructed in the proof of Lemma 6.5 into an instance that satisfies  $d_{\max} \leq u_{\min}$ . For the special class of MIN-CON-USPR instances constructed in the proof of Lemma 6.5, this transformation is strongly polynomial. Therefore, the inapproximability results of Lemma 6.5 and Theorem 6.7 also hold for the case where  $d_{\max} \leq u_{\min}$ .

### 6.4.2 Capacitated Network Design

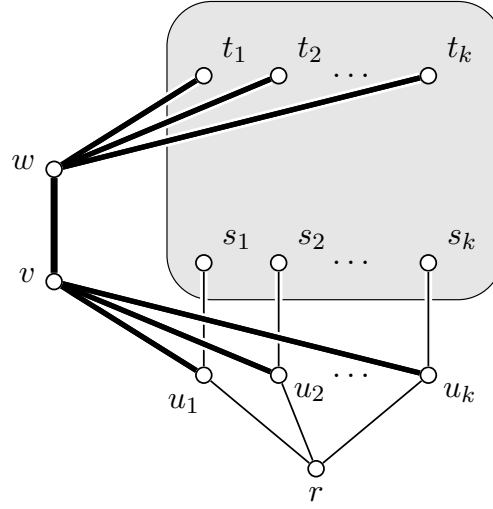
A problem that is closely related to the capacitated network design problem CAP-USPR is the GENERALIZED STEINER NETWORK problem, also known as POINT-TO-POINT CONNECTION problem: Given a (directed) graph  $D = (V, A)$  and with arc costs  $w_a \in \mathbb{Z}_+$ ,  $a \in A$ , and a set of commodities  $K \subseteq V \times V$  find a minimum cost arc set  $B \subseteq A$  such that the subgraph  $(V, B)$  contains an  $(s, t)$ -path for each  $(s, t) \in K$ .

With demand values  $d_{s,t} := 1$  for all  $(s, t) \in K$  and  $u_a := |K|$  for all  $a \in A$ , the GENERALIZED STEINER NETWORK problem reduces straightforward to the CAP-USPR problem. Inapproximability results for the directed and undirected GENERALIZED STEINER NETWORK problem carry over immediately to the corresponding CAP-USPR problem.

It was shown by Dodis and Khanna [75] that there exists no polynomial time  $\mathcal{O}(2^{\log^{1-\epsilon}|V|})$ -approximation algorithm for the directed GENERALIZED STEINER NETWORK problem with  $\epsilon > 0$ , unless  $\mathcal{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$ . This yields the strongest inapproximability threshold for the directed CAP-USPR problem currently known.

**Theorem 6.8 ([75])** *For any  $\epsilon > 0$ , the directed CAP-USPR problem is inapproximable within  $\mathcal{O}(2^{\log^{1-\epsilon}|V|})$ , unless  $\mathcal{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$ .*

For the undirected GENERALIZED STEINER NETWORK problem, Goemans and Williamson [99] devised a simple primal dual algorithm that achieves a worst case performance guarantee of  $2 - 1/|K|$ , the best known inapprox-



**Figure 6.9** Constructed CAP-USPR instance. Edges with capacity  $2M(k-1)$  and  $2Mk(k-1)$  are bold, edges with capacity 1 are thin.

imability threshold is well below this number.<sup>1)</sup> In the following, we prove a slightly stronger result for the undirected CAP-USPR problem.

**Theorem 6.9** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate the undirected CAP-USPR problem within a factor of  $2 - \epsilon$ .*

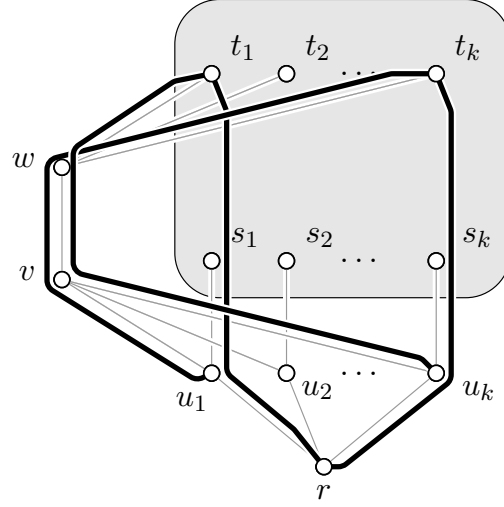
**Proof.** We present a reduction similar to the one used in the proof of Lemma 6.5.

Suppose we are given an undirected FULLY DISJOINT PATHS instance consisting of the graph  $H = (W, F)$  and the node pairs  $(s_i, t_i)$ ,  $i = 1, \dots, k$ . We may assume w.l.o.g. that there is an  $(s_i, t_i)$ -path in  $H$  for each  $i = 1, \dots, k$ , and that  $\{s_i, t_i\} \cap \{s_j, t_j\} = \emptyset$  for all  $i \neq j$ . Let  $M := \lceil (2-\epsilon)k/\epsilon \rceil + 1$ .

We construct an undirected CAP-USPR instance  $(G, u, w, K, d)$  as shown in Figure 6.9: The node set  $V$  of the graph  $G = (V, E)$  contains all nodes in  $W$ , the three nodes  $v$ ,  $w$ , and  $r$ , and a node  $u_i$  for each  $i = 1, \dots, k$ . The edge set  $E$  consists of all edges in  $F$ , one edge  $vw$ , and the edges  $ru_i$ ,  $u_i s_i$ ,  $u_i v$ , and  $wt_i$  for each  $i = 1, \dots, k$ .

The commodity set  $K$  contains two types of commodities: For each  $i = 1, \dots, k$ , we introduce a commodity  $(r, t_i)$  with demand value  $d_{r, t_i} := 1$ . For all pairs  $i, j = 1, \dots, k$  with  $i \neq j$ , we introduce a commodity  $(u_i, t_j)$  with a demand of  $d_{u_i, t_j} := 2M$ .

<sup>1)</sup> Andrews [7] shows that there is no  $\mathcal{O}(\log^{1-\epsilon}|V|)$ -approximation algorithm for the BUY-AT-BULK NETWORK DESIGN problem unless  $\mathcal{NP} \subseteq \text{ZPTIME}(n^{\text{polylog}(n)})$ . This proof can be adapted to show the same threshold for the undirected CAP-USPR problem. However, Andrews' construction inherently uses randomization and relies on the probabilistic Erdős-Sachs theorem [79]; a deterministic construction that yields the same bound is not known.



**Figure 6.10** USPR in  $G$  if fully disjoint  $(s_i, t_i)$ -paths exist in  $H$ .

The edge capacities and costs are defined as

$$u_e := \begin{cases} 2Mk(k-1), & \text{if } e = vw, \\ 2M(k-1), & \text{if } e \in \{u_i v, wt_i : i = 1, \dots, k\}, \\ 1, & \text{otherwise, and} \end{cases}$$

$$w_e := \begin{cases} M, & \text{if } e = vw, \\ 1, & \text{if } e \in \{u_i s_i : i = 1, \dots, k\}, \\ 0, & \text{otherwise.} \end{cases}$$

For any fixed  $\epsilon > 0$ , this construction is polynomial in the size of  $H$ .

First, suppose there exist fully disjoint  $(s_i, t_i)$ -paths  $P_i$  in  $H$ . Then we define the metric  $\lambda$  as

$$\lambda_e := \begin{cases} 1, & \text{if } e \in P_i \text{ for some } i \in \{1, \dots, k\} \text{ or} \\ & \text{if } e \in \{ru_i, u_i s_i, u_i v, wt_i : i = 1, \dots, k\}, \\ |A|, & \text{if } e = vw, \text{ and} \\ |A| + 2, & \text{otherwise.} \end{cases}$$

It is easy to verify that this metric  $\lambda$  induces a USPR for the commodity set  $K$ , as illustrated in Figure 6.10. For each  $i = 1, \dots, k$ , the unique shortest  $(r, t_i)$ -path is  $(r, u_i, s_i) \oplus P_i$ , and for pair  $i, j = 1, \dots, k$  with  $i \neq j$ , the unique shortest  $(u_i, t_j)$ -path is  $(u_i, v, w, t_j)$ . If the corresponding commodities are routed along these paths, the induced edge flows  $f_e$  do not exceed the given capacities  $u_e$ ,  $e \in E$ . Hence, the metric  $\lambda$  and the capacity multipliers  $z_e = 1$  for all  $e \in E$  form a feasible solution for the constructed CAP-USPR

instance. The total capacity installation cost for this solution is

$$\sum_{e \in E} w_e z_e = \sum_{i=1}^k w_{u_i s_i} + w_{vw} = k + M.$$

Now, suppose there is no set of fully disjoint  $(s_i, t_i)$ -paths in  $H$ . Let  $\lambda \in \mathbb{Z}_+^E$  be an arbitrary metric that defines a USPR for the commodities  $K$ . Analogous to the proof of Lemma 6.5, there are two possible cases: Either some commodity  $(u_i, t_j)$ ,  $i \neq j$ , is routed across an edge  $u_l s_l$ , or all commodities  $(u_i, t_j)$ ,  $i \neq j$ , and (at least) one commodity  $(r, t_i)$ ,  $i \in \{1, \dots, k\}$ , are routed together across the edge  $vw$ .

In the first case, a commodity  $(u_i, t_j)$  with demand value  $d_{u_i, t_j} = 2M$  is routed across an edge  $u_l s_l$  with  $u_{u_l s_l} = 1$ . Then the capacity multiplier  $z_{u_l s_l}$  must be at least  $2M$  and, therefore, the cost of this solution is no less than  $z_{u_l s_l} w_{u_l s_l} = 2M$ .

In the second case, all  $k(k-1)$  commodities  $(u_i, t_j)$ ,  $i \neq j$ , with  $d_{u_i, t_j} = 2M$  and (at least) one commodity  $(r, t_l)$  with  $d_{r, t_l} = 1$  are routed across the edge  $vw$  with  $u_{vw} = 2Mk(k-1)$ . Then the capacity multiplier  $z_{vw}$  must be at least two, which also yields a total solution cost of at least  $2M$ .

Hence, if there are no fully disjoint  $(s_i, t_i)$ -paths in  $H$ , then any feasible solution of the constructed CAP-USPR instance has a cost of at least  $2M$ .

Since it is  $\mathcal{NP}$ -hard to decide whether fully disjoint  $(s_i, t_i)$ -paths exist in  $H$ , it is also  $\mathcal{NP}$ -hard to approximate the optimal solution of the constructed undirected CAP-USPR instance within a factor strictly less than  $2M/(k+M) > 2 - \epsilon$ , as claimed.  $\square$

Theorem 6.9 also holds for the directed CAP-USPR problem, but the stronger general inapproximability threshold for the directed case follows from Theorem 6.8. With a construction similar to the one presented in the proof of Theorem 6.9 one can show that the directed CAP-USPR problem remains hard to approximate within  $2 - \epsilon$  even if  $|K| = 2$ . For the special cases where the underlying digraph is a cycle or a bidirected ring, a variant of the reduction used in Theorem 6.1 yields a constant inapproximability threshold of  $4/3$  for the both the undirected and the directed CAP-USPR problem, respectively.

### 6.4.3 Fixed Charge Network Design

Intuitively, the fixed charge problem FC-USPR is harder than the CAP-USPR problem, where the installation of arbitrarily large arc capacities is allowed. For any given CAP-USPR instance, there exists a feasible solution which can be easily found (provided that  $D$  contains an  $(s, t)$ -path for each  $(s, t) \in K$ ). For FC-USPR, on the other hand, already the task of finding some feasible solution is  $\mathcal{NP}$ -hard, cf. Theorem 6.1. If we were given an FC-USPR instance with cost one for some arcs and prohibitively large costs for

all others, then the core of the problem is to find a USPR in the subgraph induced by the edges of cost one. As this is an  $\mathcal{NP}$ -complete problem, we cannot expect to find a solution of FC-USPR with a reasonable quality guarantee in polynomial time. In the following, we prove this intuition.

**Theorem 6.10** *FC-USPR is  $\mathcal{NPO}$ -complete.*

**Proof.** We present an approximation preserving reduction, to be more precisely a  $\mathcal{PTAS}$ -reduction [10, 65], from the MIN-WEIGHT-SAT problem.

The MIN-WEIGHT-SAT problem is defined as follows: Given a set  $X$  of boolean variables, a collection  $C$  of disjunctive clauses of at most three literals per clause, and a non-negative integer weight for each variable in  $X$ , the aim is to find a truth assignment for  $X$  that satisfies all clauses in  $C$  and minimizes the sum of the weights of the *true* variables. In MIN-WEIGHT-SAT(3), each variable occurs at most three times in total and at least once as a negated and once as an unnegated literal. It was shown by Orponen and Mannila [150] that MIN-WEIGHT-SAT is  $\mathcal{NPO}$ -complete. As for the unweighted satisfiability problem, the restricted problem MIN-WEIGHT-SAT(3) remains  $\mathcal{NPO}$ -complete. The restriction to 3 occurrences of each boolean variable is not necessary to prove Theorem 6.10, but it allows us to use a simpler reduction where all demand values and capacities are either 1 or 2.

Suppose we are given a MIN-WEIGHT-SAT(3) instance consisting of the boolean variables  $x_1, \dots, x_n$  with nonnegative weights  $w_i \in \mathbb{Z}_+$ ,  $1 \leq i \leq n$ , and the clauses  $C_1, \dots, C_m$ . We construct an instance  $(D, u, w, K, d)$  of FC-USPR, such that any truth assignment for the MIN-WEIGHT-SAT(3) instance corresponds to a solution of the FC-USPR instance whose cost is equal to the weight of the truth assignment.

The digraph  $D = (V, A)$  contains the  $6n + 2m$  nodes

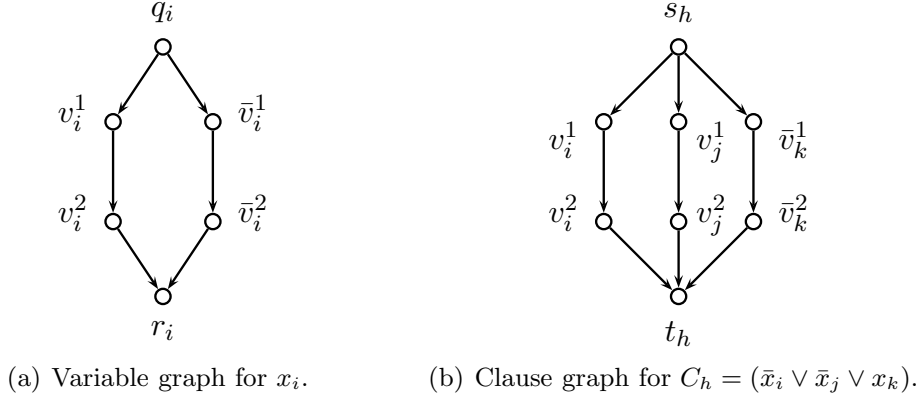
$$V := \{q_i, v_i^1, v_i^2, \bar{v}_i^1, \bar{v}_i^2, r_i : i = 1, \dots, n\} \cup \{s_h, t_h : h = 1, \dots, m\} .$$

Among these nodes, we introduce the arcs

$$A_x := \{(q_i, v_i^1), (v_i^1, v_i^2), (v_i^2, r_i), (q_i, \bar{v}_i^1), (\bar{v}_i^1, \bar{v}_i^2), (\bar{v}_i^2, r_i) : i = 1, \dots, n\} , \text{ and}$$

$$A_C := \{(s_h, \bar{v}_i^1), (\bar{v}_i^2, t_h) : i = 1, \dots, n \text{ and } h = 1, \dots, m \text{ such that } x_i \text{ appears unnegated in } C_h\} \\ \cup \{(s_h, v_i^1), (v_i^2, t_h) : i = 1, \dots, n \text{ and } h = 1, \dots, m \text{ such that } x_i \text{ appears negated in } C_h\} .$$

For each boolean variable  $x_i$ , the nodes indexed by  $i$  form a *variable subgraph* as shown in Figure 6.11(a). For each clause  $C_h$ , the nodes  $s_h, t_h$ , and either



**Figure 6.11** Constructed FC-USPR instance.

the nodes  $v_i^1, v_i^2$  or the nodes  $\bar{v}_i^1, \bar{v}_i^2$  with index  $i$  such that  $x_i$  occurs in  $C_h$  form a *clause subgraph* as shown in Figure 6.11(b).

The arc capacities and costs are defined as

$$u_a := \begin{cases} 2, & \text{if } a \in A_x, \\ 1, & \text{if } a \in A_C, \end{cases}$$

and

$$w_a := \begin{cases} w_i, & \text{if } a = (q_i, v_i^1), i \in \{1 \dots, n\}, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

The commodity set  $K$  consists of a commodity  $(q_i, r_i)$  with a demand value of  $d_{q_i, r_i} = 2$  for each  $i = 1, \dots, n$ , and of a commodity  $(s_h, t_h)$  with a demand  $d_{s_h, t_h} = 1$  for each  $h = 1, \dots, m$ .

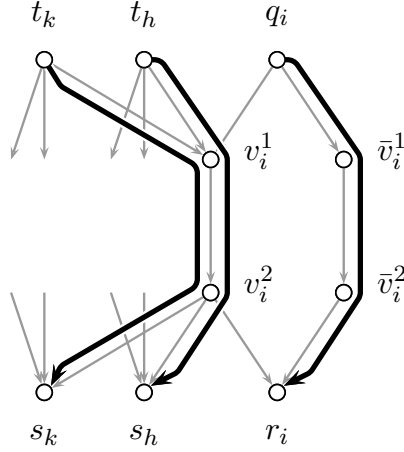
Clearly, this construction is polynomial in the size of the given MIN-WEIGHT-SAT(3) instance.

In the first part of the proof, we show that, for each truth assignment that satisfies all clauses of the MIN-WEIGHT-SAT(3) instance, there exists a corresponding feasible solution of the constructed FC-USPR instance. So, let  $x \in \{true, false\}^n$  be such a truth assignment.

For each clause  $C_h$ , at least one literal evaluates to *true*. The index of the corresponding binary variable is denoted by  $i(h) \in \{1, \dots, n\}$ . If more than one literals evaluate to true in  $C_h$ , then  $i(h)$  w.l.o.g. denotes the lexicographically first one. We define the corresponding metric  $\lambda = \lambda(x) \in \mathbb{Z}_+^A$  as

$$\lambda_a := \begin{cases} 1, & \text{if } a \in \bigcup_{i: x_i = true} \{(q_i, v_i^1), (v_i^1, v_i^2), (v_i^2, r_i)\} \text{ or} \\ & \text{if } a \in \bigcup_{i: x_i = false} \{(q_i, \bar{v}_i^1), (\bar{v}_i^1, \bar{v}_i^2), (\bar{v}_i^2, r_i)\} \text{ or} \\ & \text{if } a \in \bigcup_{h: C_h = (\bar{x}_i(h) \vee \dots)} \{(s_h, v_{i(h)}^1), (v_{i(h)}^2, t_h)\} \text{ or} \\ & \text{if } a \in \bigcup_{h: C_h = (x_i(h) \vee \dots)} \{(s_h, \bar{v}_{i(h)}^1), (\bar{v}_{i(h)}^2, t_h)\}, \text{ and} \\ 2, & \text{otherwise.} \end{cases}$$





**Figure 6.12** Partial routing in  $D$  corresponding to a truth assignment with  $x_i = \text{false}$  and clauses  $C_h = (\bar{x}_i \vee \dots)$  and  $C_k = (\bar{x}_i \vee \dots)$ .

This metric  $\lambda$  defines a USPR for the commodities in  $K$ . If the boolean variable  $x_i$  is *true*, then we route commodity  $(q_i, r_i)$  on path  $P_i^+ := (q_i, v_i^1, v_i^2, r_i)$ , otherwise on path  $P_i^- := (q_i, \bar{v}_i^1, \bar{v}_i^2, r_i)$ . If the boolean variable  $x_{i(h)}$  occurs negated in  $C_h$ , then we route commodity  $(s_h, t_h)$  along the path  $(s_h, v_i^1, v_i^2, t_h)$ . (Note that  $\bar{x}_{i(h)}$  then is *true* by definition of  $i(h)$ .) Otherwise, if the boolean variable  $x_{i(h)}$  occurs unnegated (and evaluates to *true*) in clause  $C_h$ , then commodity  $(s_h, t_h)$  is routed via path  $(s_h, \bar{v}_i^1, \bar{v}_i^2, t_h)$ . Figure 6.12 illustrates this routing.

Since each variable  $x_i$  occurs in at most two clauses negated and in at most two clauses unnegated in the given MIN-WEIGHT-SAT(3) instance, any arc  $(v_i^1, v_i^2)$  or  $(\bar{v}_i^1, \bar{v}_i^2)$  is contained in at most two commodity routing paths. All other arcs  $a \in A$  are contained in at most one shortest path with respect to  $\lambda$ . Hence, the flows that are induced by the corresponding USPR satisfy the given capacities.

Let  $B \subseteq A$  be the set of arcs contained in the induced routing paths. Then  $(B, \lambda)$  defines a feasible solution of the FC-USPR instance. Clearly,  $(q_i, v_i^1) \in B$  if and only if the boolean variable  $x_i$  is *true*. Hence, we have

$$\sum_{a \in B} w_a = \sum_{i: x_i = \text{true}} w_i ,$$

i.e., the weight of the truth assignment  $x$  is the same as the cost of its corresponding FC-USPR solution  $(B, \lambda)$ . This immediately implies

$$w(\lambda^{\text{opt}}) \leq w(x^{\text{opt}}) . \quad (6.2)$$

In the second part of the proof, we show that any feasible solution of the constructed FC-USPR instance defines a truth assignment satisfying all clauses. Let  $(B, \lambda)$  be such a feasible solution. First, observe that each commodity  $(q_i, r_i)$ ,  $i \in \{0, \dots, n\}$ , is routed either on path  $P_i^+$  or on path  $P_i^-$ . Any other  $(q_i, r_i)$ -path contains some arc  $a \in A_H$ , whose capacity

$u_a = 1$  is insufficient to accommodate the demand  $d_{q_i, r_i} = 2$  of commodity  $(q_i, r_i)$ . Thus, we can define the truth assignment  $x = x(\lambda) \in \{true, false\}^n$  as

$$x_i := \begin{cases} true, & \text{if } P_i^+ \text{ is the (unique) shortest } (q_i, r_i)\text{-path w.r.t. } \lambda, \\ false, & \text{otherwise.} \end{cases}$$

If the shortest  $(s_h, t_h)$ -path w.r.t.  $\lambda$  contains the vertex  $v_i^1$ , then commodity  $(q_i, r_i)$  must be routed on the path  $P_i^-$  in the USPR defined by  $\lambda$ . Otherwise, the capacity of arc  $(v_i^1, v_i^2)$  would be violated. Analogously, commodity  $(q_i, r_i)$  must be routed via  $P_i^+$  if  $\bar{v}_i^1$  is contained in the routing path of some commodity  $(s_h, t_h)$ . According to our construction, this implies that the corresponding clause  $C_h$  evaluates to *true* for the truth assignment  $x$  defined by the given routing. Hence, the constructed truth assignment  $x$  satisfies all clauses.

The arc set  $B$  of the given FC-USPR solution contains all arcs that belong to some of the routing paths. For each boolean variable  $x_i$  that was set to *true* in the constructed truth assignment  $x$ , the arc  $(q_i, v_i^1)$  is contained in the routing path for commodity  $(q_i, r_i)$  and therefore also belongs to the set  $B$ . Hence, we have

$$\sum_{a \in B} w_a \geq \sum_{i: x_i = true} w_i. \quad (6.3)$$

Together (6.2) and (6.3) imply that any  $\epsilon$ -approximate solution of the constructed FC-USPR instance corresponds to an  $\epsilon$ -approximate solution of the original MIN-WEIGHT-SAT(3) instance. Thus, the given reduction is a  $\mathcal{PTAS}$ -reduction from MIN-WEIGHT-SAT(3) to FC-USPR. As MIN-WEIGHT-SAT(3) is  $\mathcal{NPO}$ -complete, so is FC-USPR.  $\square$

Theorem 6.10 immediately yields the following corollary.

**Corollary 6.11** *For any  $\epsilon > 0$ , it is  $\mathcal{NP}$ -hard to approximate FC-USPR within a factor of  $2^{\langle I \rangle^{1-\epsilon}}$ , where  $\langle I \rangle$  is the encoding size of the FC-USPR instance  $I = (D, u, w, K, d)$ .*

In the proof of Theorem 6.10, we only used the requirement that all commodities are routed unsplitted. Our construction implicitly guarantees the existence of a compatible metric for every unsplittable flow routing that satisfies the given capacities. Hence, Theorem 6.10 and Corollary 6.11 also hold for the fixed charge network design problem with unsplittable flow routing instead of unsplittable shortest path routing.

Theorem 6.10 and Corollary 6.11 also hold if the underlying digraph is a bidirected ring (or a cycle in the undirected case), which can be easily shown with a construction similar to that in the proof of Theorem 6.1.

Finally, note that any  $\mathcal{NPO}$  problem can be approximated within a factor of  $\mathcal{O}(2^{(1)^\epsilon})$  for some  $\epsilon > 0$  (i.e., it belongs to the class  $\text{exp-APX}$ ) if at least one feasible solution can be computed in polynomial time. Because for FC-USPR already the problem of finding a feasible solution is  $\mathcal{NP}$ -hard, this problem does not belong to  $\text{exp-APX}$ , unless  $\mathcal{P} = \mathcal{NP}$ .

## 6.5 General Approximation Algorithms

In this section, we present polynomial time approximation algorithms for the minimum congestion problem MIN-CON-USPR and for the capacitated network design problem CAP-USPR that are applicable for general underlying digraphs. The fixed charge network design problem FC-USPR is not approximable within any reasonable quality guarantee, unless  $\mathcal{P} = \mathcal{NP}$ .

### 6.5.1 Congestion Minimization

By Theorem 6.1, MIN-CON-USPR is  $\mathcal{NP}$ -hard to approximate within a factor of  $\mathcal{O}(|V|^{1-\epsilon})$ . In the following, we show how to compute  $\min\{|K|, |A|\}$ -approximate solutions.

We begin by showing how to approximate MIN-CON-USPR within a factor of  $|K|$ . Note that, in contrast to the GENERALIZED STEINER NETWORK problem, this is not trivial: We have to ensure that there exist a compatible metric for the chosen routing paths, i.e., a metric such that each routing path is the unique shortest path between its terminals.

**Definition 6.12** *For each path  $P \in \mathcal{P}$ , let  $u_{\min}(P) := \min\{u_a : a \in P\}$  be the **thickness** of  $P$ .*

An obvious optimal solution for a USPR instance with only one commodity  $(s, t)$  is to route  $(s, t)$  via an  $(s, t)$ -path  $P_{(s,t)}^*$  of maximum thickness, i.e., a path with  $u_{\min}(P_{(s,t)}^*) = \max\{u_{\min}(P) : P \in \mathcal{P}(s, t)\}$ . Choosing a maximum thickness  $(s, t)$ -path  $P_{(s,t)}^*$  for each commodity  $(s, t)$  therefore yields an unsplittable flow routing whose congestion is at most  $|K|$  times the congestion of an optimal unsplittable flow routing in the multicommodity case.

Note that maximum thickness paths are not necessarily unique. Furthermore, it also may be impossible to enforce uniqueness by a small perturbation of the capacities. In order to guarantee the existence of a compatible metric for the chosen paths, we need to consider *all* capacities on the paths instead of only the bottleneck capacity.

**Definition 6.13** *For each path  $P \in \mathcal{P}$ , the **capacity pattern**  $u_{\text{seq}}(P)$  of  $P$  is the non-decreasingly sorted sequence of its arc capacities, i.e.,  $u_{\text{seq}}(P) := (u_{a_1}, \dots, u_{a_{|P|}})$  with  $a_i \in P$ ,  $a_i \neq a_j$ , and  $u_{a_i} \leq u_{a_{i+1}}$ .*

The lexicographic order on the capacity patterns defines a prefix-monotone total order on the paths. In contrast to the order by path thickness, ties in the  $u_{seq}$ -order can be broken consistently by an arbitrarily small perturbation of the capacities (or via a secondary lexicographical ordering of the paths according to their arc numbers), for example. It is not difficult to verify that, if each path  $P_{(s,t)}^*$  is the unique  $u_{seq}$ -maximum  $(s, t)$ -path with respect to (a perturbation of) the arc capacities  $u$ , then there exists a metric  $\lambda \in \mathbb{Z}_+^A$  such that each  $P_{(s,t)}^*$  is a unique shortest  $(s, t)$ -path w.r.t.  $\lambda$ . This leads to the simple approximation algorithm **ThickestPath** shown below.

**Theorem 6.14** *ThickestPath is an  $|K|$ -approximation algorithm for MIN-CON-USPR.*

**Proof.** Obviously, any  $u_{seq}$ -maximal  $(s, t)$ -path is a maximum thickness  $(s, t)$ -path. As we route each commodity  $(s, t) \in K$  on the  $u_{seq}$ -maximal path with respect to (a perturbation of) the capacities  $u_a$ , the paths  $P_{(s,t)}^*$  hence form a routing with congestion at most  $|K|$  times the congestion of an optimal solution.

As the lexicographic order on the  $u_{seq}$ -sequences defines a prefix-monotone total order on the paths, the  $u_{seq}$ -maximal  $(s, t)$ -paths can be found in polynomial time using a standard labeling algorithm. If the ties in the  $u_{seq}$ -order are broken consistently (by an appropriate perturbation), these paths form an unsplittable shortest path path routing. A compatible metric then can be computed in polynomial time using the linear programming and rounding approaches discussed in Chapter 4.  $\square$

Our second approximation algorithm, whose performance guarantee is independent of the number of commodities, is based on the multicommodity flow relaxation of the MIN-CON-USPR problem. In a fractional multicommodity flow (MCF) routing, the demand of each commodity  $(s, t)$  may be

---

**Algorithm 6.1** ThickestPath

---

**Input:** Digraph  $D = (V, A)$  with capacities  $u_a \in \mathbb{Z}_+$  for all  $a \in A$  and a commodity set  $K \subseteq V \times V$  with demands  $d_{s,t} \in \mathbb{Z}_+$  for all  $(s, t) \in K$ .

**Output:** A valid metric  $\lambda \in \mathbb{Z}_+^A$  for the commodity set  $K$ .

1. Compute the  $u_{seq}$ -maximal  $(s, t)$ -path  $P_{(s,t)}^*$  for each commodity  $(s, t) \in K$  with respect to (a perturbation) of the capacities  $u_a$ ,  $a \in A$ .
  2. Find a metric  $\lambda \in \mathbb{Z}_+^A$  such that for each  $(s, t) \in K$  the path  $P_{(s,t)}^*$  is the unique shortest  $(s, t)$ -path w.r.t.  $\lambda$ . (Possible algorithms are discussed in Chapter 4).
  3. Return  $\lambda$ .
-

distributed arbitrarily among the paths  $\mathcal{P}(s, t)$ . Such a routing can be expressed as an assignment  $x : \mathcal{P}(K) \rightarrow [0, 1]$ , where each  $x_P$  denotes the fraction of the demand  $d_{s,t}$  that is sent along  $P \in \mathcal{P}(s, t)$ . The problem of finding an MCF routing of minimal congestion can be formulated as a linear program as follows:

$$\begin{aligned}
 & \min L && \text{(CON-LP)} \\
 & \sum_{P \in \mathcal{P}(s,t)} x_P = 1 && \forall (s, t) \in K \\
 & \sum_{(s,t) \in K} \sum_{P \in \mathcal{P}(s,t): a \in P} d_{s,t} \cdot x_P \leq L \cdot u_a && \forall a \in A \\
 & L \geq 1 \\
 & 0 \leq x_P \leq 1 && \forall (s, t) \in K, P \in \mathcal{P}(s, t)
 \end{aligned} \tag{6.4}$$

Using column generation techniques, (CON-LP) can be solved in polynomial time w.r.t. to the size of the given problem instance, even though it contains exponentially many path variables. Let  $(x^*, L^*)$  be an optimal solution of (CON-LP). Clearly,  $L^*$  is a lower bound for the minimum congestion that can be obtained with an unsplittable shortest path routing. The total flow across an arc  $a \in A$  in the corresponding multicommodity flow routing is

$$f_a(x^*) := \sum_{(s,t) \in K} \sum_{P \in \mathcal{P}(s,t): a \in P} d_{s,t} x_P^* .$$

Let  $\pi_a^* \in \mathbb{R}_+$ ,  $a \in A$ , be the optimal dual variables corresponding to the constraints (6.4) in (CON-LP). It follows from LP duality that all paths  $P$  with  $x_P^* > 0$  are shortest paths between their respective terminal nodes with respect to the metric  $\pi^*$ . However, these paths are not necessarily *unique* shortest paths with respect to  $\pi^*$  and, furthermore, not all shortest  $(s, t)$ -paths with respect to  $\pi^*$  for a commodity  $(s, t)$  do carry a positive flow. The idea of our second algorithm **PenalizeSmallLinks**, which is illustrated on the following page, is to perturb the metric given by the optimal dual variables  $\pi_a^*$  such that each commodity's shortest path is unique.

**Theorem 6.15** *PenalizeSmallLinks is an  $|A|$ -approximation algorithm for MIN-CON-USPR.*

**Proof.** It is easy to see that there exist an integer-valued metric  $\lambda' \in \mathbb{Z}_+^A$  that induces exactly the same shortest paths as the given fractional metric  $\pi^* \in \mathbb{R}_+^A$  (whether or not uniqueness is an issue). One such integer-valued metric can be found by considering the optimal basis of (CON-LP) computed (implicitly) in Step 1 of the algorithm and then scale  $\pi^*$  according to Cramer's rule, for example.

Given an integer valued metric  $\lambda' \in \mathbb{Z}_+^A$  and an arbitrary arc numbering  $idx : A \rightarrow \{1, \dots, |A|\}$ , the shortest  $(s, t)$ -path in  $D$  with respect to the

'perturbed' metric  $\lambda_a := 2^{|A|+1} \cdot \lambda'_a + 2^{idx(a)}$  is unique for every pair of connected nodes  $s, t$ . Hence, the metric  $\lambda$  defined in Step 4 of algorithm `PenalizeSmallLinks` is valid also for the given commodity set  $K$ . It remains to show that the congestion induced of this USPR is at most  $|A|$  times the congestion of the optimum MCF routing.

Let  $f_a(x^*)$  and  $f_a(\lambda)$  denote the arc flows induced by an optimal solution  $(x^*, L^*)$  of (CON-LP) and by the USPR defined by the metric  $\lambda$ , respectively. Compared to the optimal MCF flow, an arc  $a$  receives additional traffic in the USPR routing only from those arcs  $a'$  with  $idx(a') > idx(a)$ . Hence, we have for each  $a \in A$

$$f_a(\lambda) \leq \sum_{a': idx(a') \geq idx(a)} f_{a'}(x^*) \leq \sum_{a': f_{a'}(x^*) \leq f_a(x^*)} f_{a'}(x^*) \leq |A| \cdot f_a(x^*) .$$

Consequently,  $L^* \leq \max_{a \in A} f_a(\lambda)/u_a \leq \max_{a \in A} |A| \cdot f_a(x^*)/u_a = |A| \cdot L^*$ .  $\square$

We have already seen the proof of Proposition 6.2 that a ratio of  $\Omega(|A|)$  between the optimal congestion values for USPR and MCF routings may be attained. Together, Theorems 6.14 and 6.15 yield the following corollary.

**Corollary 6.16** *MIN-CON-USPR is polynomially approximable within a factor of  $\min\{|K|, |A|\}$ .*

Note that the routing obtained by rounding the optimal fractional multi-commodity flow routing  $x^*$  is not necessarily a valid USPR. In the following example, all basic optimal solutions of (CON-LP) are integer and thus form unsplittable flow routings, but none of these routings is a USPR.

**Example 6.1** Consider the bidirected ring  $D = (V, A)$  consisting of the nodes  $V = \{v_1, \dots, v_4\}$  and the arcs  $A = \{(v_1, v_2), (v_2, v_1), \dots, (v_1, v_4)\}$ .

---

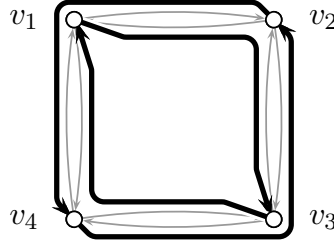
**Algorithm 6.2** `PenalizeSmallLinks`

---

**Input:** Digraph  $D = (V, A)$  with capacities  $u_a \in \mathbb{Z}_+$  for all  $a \in A$  and a commodity set  $K \subseteq V \times V$  with demands  $d_{s,t} \in \mathbb{Z}_+$  for all  $(s, t) \in K$ .

**Output:** A valid metric  $\lambda \in \mathbb{Z}_+^A$  for the commodity set  $K$ .

1. Compute an optimal solution  $(x^*, L^*)$  of (CON-LP).  
Let  $\pi_a^*$ ,  $a \in A$ , be the optimal dual variables for (6.4).
  2. Find *integer* arc lengths  $\lambda'_a \geq 1$  that induce the same shortest paths as  $\pi_a^*$ . (In particular, such that all paths  $P$  with  $x_P^* > 0$  are shortest paths w.r.t.  $\lambda$ .)
  3. Number the arcs by  $idx : A \rightarrow \{1, \dots, |A|\}$  in order of non-increasing flow values  $f_a(x^*)$ .
  4. Return the metric  $\lambda_a := 2^{|A|+1} \cdot \lambda'_a + 2^{idx(a)}$ .
-



**Figure 6.13** Optimal basic solution of (CON-LP) for Example 6.5.1.

Let  $u_a = 1$  for all  $a \in A$ , and consider the four commodities  $K := \{(v_1, v_3), (v_3, v_1), (v_2, v_4), (v_4, v_2)\}$  with demand values  $d_{s,t} = 1$  for all  $(s, t) \in K$ .

For this instance of MIN-CON-USPR, all basic optimal solutions of (CON-LP) are integer and up to symmetry correspond to the routing  $x_P = 1$  for all  $P \in S := \{(v_1, v_2, v_3), (v_3, v_4, v_1), (v_2, v_1, v_4), (v_4, v_3, v_2)\}$  and  $x_P = 0$  for all  $P \notin S$ , which is illustrated in Figure 6.13. This routing, however, is not a USPR. One easily verifies that the linear system (4.3)–(4.4) is infeasible for the corresponding path set  $S$ . So, by Theorem 4.4, there exists no compatible metric for  $S$ . In fact, any feasible USPR for commodities  $K$  induces a flow of at least two on at least one arc of the bidirected ring.  $\triangle$

Algorithms `ThickestPath` and `PenalizeSmallLinks` carry over straightforward to the undirected version of MIN-CON-USPR, where they have worst case approximation ratios of  $|K|$  and  $|E|$ , respectively. In Section 6.6, we show that `PenalizeSmallLinks` achieves a constant factor-2 approximation guarantee in the special case where the underlying graph is an undirected cycle.

### 6.5.2 Capacitated Network Design

As many other capacitated network design problems, the CAP-USPR problem seems to be very hard to approximate. No algorithms with non-trivial quality guarantees are known for general arc capacities and arbitrary commodities.<sup>2)</sup> However, the uniform and the single-source version of the problem can be approximated within reasonable bounds.

We say that a CAP-USPR problem instance is **uniform** if all arc (or edge) capacities are equal, i.e.,  $u_a = u$  for all  $a \in A$ . For the uniform CAP-USPR problem with only one commodity  $(s, t)$ , an optimal solution is given by routing the commodity along some shortest  $(s, t)$ -path  $P^*$  with respect

<sup>2)</sup>The  $2^{\mathcal{O}(\sqrt{\log |V| \log \log |V|}) \log d_{\max}}$ -approximation algorithm of Charikar and Karagiuzova [56] for the BUY-AT-BULK NETWORK DESIGN problem does not carry over to the CAP-USPR problem, as the routings computed by this algorithm do not necessarily form unsplittable shortest path routings.

to the arc costs  $w$  and installing  $\lceil d_{s,t}/u \rceil$  many capacity units on the arcs of  $P^*$ . For the uniform CAP-USPR problem with multiple commodities, sending each commodity  $(s, t) \in K$  along a shortest  $(s, t)$ -path  $P_{(s,t)}^*$  with respect to (an appropriate perturbation of) the arc costs  $w$  and installing  $\lceil \sum_{(s,t): a \in P_{(s,t)}^*} d_{s,t}/u \rceil$  many capacity units on each arc  $a \in A$  thus trivially yields an  $|K|$ -approximate solution.

**Proposition 6.17** *The CAP-USPR problem with uniform capacities is approximable within a factor of  $|K|$ .*

Clearly, Proposition 6.17 holds for both the directed and the undirected problem version. For the non-uniform CAP-USPR problem, however, routing each commodity individually along an optimal path for its corresponding single commodity problem does not necessarily yield a valid unsplittable shortest path routing for the entire commodity set.

For the *undirected* uniform CAP-USPR problem, the trivial bound of Proposition 6.17 can be improved to a factor of only  $\mathcal{O}(\log |V|)$  using a probabilistic approximation of the arc costs by dominating tree metrics.<sup>3)</sup> Given an undirected graph  $G = (V, E)$  and a metric  $\lambda \in \mathbb{R}_+^E$ , we denote by  $\text{dist}_\lambda(s, t)$  the distance between  $s$  and  $t$  with respect to  $\lambda$  in  $G$ . A metric  $\lambda \in \mathbb{R}_+^E$  is said to be a **tree metric**, if there exists a tree  $T$  in  $G$  such that all shortest paths with respect to  $\lambda$  are fully contained in  $T$ . A metric  $\lambda$  **dominates** another metric  $\mu$  if  $\text{dist}_\lambda(s, t) \geq \text{dist}_\mu(s, t)$  for all  $(s, t) \in V^{(2)}$ . The **stretch** of a dominating metric  $\lambda$  with respect to another metric  $\mu$  is

$$\text{stretch}(\lambda, \mu) := \max \{ \text{dist}_\lambda(s, t) / \text{dist}_\mu(s, t) : (s, t) \in V^{(2)} \}.$$

Bartal [19] showed that any metric in an undirected graph can be probabilistically approximated by a distribution over dominating tree metrics such that the expected stretch is  $\mathcal{O}(\log^2 |V|)$ . This result was later improved by Fakcharoenphol et al. [84] to an expected stretch of only  $\mathcal{O}(\log |V|)$ . Charikar et al. [57] showed how to de-randomize this probabilistic approximation, i.e., how to approximate a metric with a distribution over only polynomially many tree metrics.

Based on these results, Awerbuch and Azar [11] proposed an approximation algorithm for the undirected uniform BUY-AT-BULK NETWORK DESIGN problem. As all solutions computed by this algorithm are trees, it carries over directly to the undirected uniform CAP-USPR problem. The de-randomized version of this algorithm basically works as follows:

First, we compute polynomially many tree metrics  $\lambda^i \in \mathbb{R}_+^E$ ,  $i \in I$ , that probabilistically approximate the given arc costs  $w$ . For each tree metric  $\lambda^i$ ,

---

<sup>3)</sup> Alternatively, also the technique of approximating the underlying graph by a so-called light-weight distance-preserving spanner (with respect to the edge costs) can be applied to obtain a  $\mathcal{O}(\log |V|)$ -approximation algorithm for the undirected uniform CAP-USPR problem, see Mansur and Peleg [137].



we then compute the cost of the solution defined by that metric, that is, by routing each commodity  $(s, t)$  along a shortest path  $P_{(s,t)}^*(\lambda^i)$  with respect to  $\lambda^i$  and installing  $\lceil \sum_{(s,t): e \in P_{(s,t)}^*(\lambda^i)} d_{s,t}/u \rceil$  many capacity units on each edge  $e \in E$ . Clearly, each such solution is feasible for the given uniform CAP-USPR problem. At the end, we return the best of these  $|I|$  solutions.

The performance guarantee of  $\mathcal{O}(\log |V|)$  follows straightforward from [11] and [84].

**Theorem 6.18 (Awerbuch and Azar[11], Fakcharoenphol et al.[84])**  
*The undirected uniform CAP-USPR problem is approximable within a factor of  $\mathcal{O}(\log |V|)$ .*

Note that in planar graphs any metric can be probabilistically approximated by polynomially many tree metrics with only constant expected stretch. Hence, Awerbuch and Azar's algorithm yields a constant worst-case guarantee for the uniform CAP-USPR problem on undirected planar graphs.

For the non-uniform and for the directed CAP-USPR problem, the technique of using probabilistic approximations by tree metrics utterly fails.

Another interesting variant of the CAP-USPR problem is the **single-source** version, where all commodities share the same source terminal (or, equivalently, share the same destination). Single-source network design problems have been considered in the literature for various capacity and routing paradigms. Most proposed solution techniques enforce that the routing paths form a tree and — as any tree routing is an unsplittable shortest path routing — can be applied directly for the single-source CAP-USPR problem.

A straightforward approach to compute a solution for the single-source problem is to iteratively assign to each commodity the cheapest path such that the new path together with the already assigned paths forms an arborescence. If the commodities are considered in order of decreasing demands, this simple Greedy algorithm achieves a worst-case guarantee of  $|K|$ . Clearly, this approach works for both the directed and the undirected single-source problem.

**Proposition 6.19** *The single-source CAP-USPR problem is approximable within a factor of  $|K|$  in general.*

For the directed single-source CAP-USPR problem, no better approximation algorithm is known.

The *undirected* single-source problem can also be solved using one of the algorithms proposed by Guha et al. [108], Gupta et al. [110], Talwar [182], or Meyerson et al. [140]. These algorithms achieve a constant approximation ratio for the uniform undirected single-source problem and an  $\mathcal{O}(\log |K|)$ -approximation ratio for the non-uniform undirected single-source problem.

None of these algorithms was specifically designed for the CAP-USPR problem, but they all produce tree routings and thus can be applied straightforward.

## 6.6 Special Cases

In the following, we present specialized algorithms that achieve constant factor approximation guarantees for MIN-CON-USPR and CAP-USPR in the special cases where the underlying graph is a bidirected ring or an undirected cycle. As mentioned above, FC-USPR remains  $\mathcal{NPO}$ -complete even in these special cases.

### 6.6.1 Min-Con-USPR on an Undirected Cycle

Algorithm `PenalizeSmallLinks` presented in Section 6.5 carries over straightforward to the undirected version of MIN-CON-USPR. In the special case where the underlying graph  $G = (V, E)$  is an undirected cycle, there are only two possible routing paths for each commodity. For any edge  $e \in E$ , exactly one of these two paths contains  $e$ . Hence, perturbing the length of only one minimum flow edge  $e_{\min} := \arg \min f_e(x^*)$  suffices to ensure that all shortest paths are unique. This immediately yields the following theorem.

**Theorem 6.20** *Algorithm `PenalizeSmallLinks` achieves a worst case approximation ratio of 2 if the underlying graph is an undirected cycle.*

**Proof.** Analogous to the proof of Theorem 6.15.  $\square$

Theorem 6.20 extends straightforward to the case where all blocks of the underlying undirected graph are cycles.

Cosares and Saniee [63] and Schrijver et al. [177] proposed approximation algorithms for the undirected RING LOADING problem, which is equivalent to the minimum congestion unsplittable flow problem on a cycle with unit capacities, that are based on rounding the optimal solution of (CON-LP). The same approach also works for the undirected MIN-CON-USPR problem, only the rounding procedure must be slightly adapted in order to guarantee that the resulting paths form an unsplittable shortest paths routing. For the general (i.e., non-uniform) MIN-CON-USPR problem on undirected cycles, the adapted algorithms also achieve a worst-case approximation ratio of 2.

### 6.6.2 Cap-USPR on an Undirected Cycle

Let  $G = (V, E)$  be an undirected cycle with edge capacities  $u_e \in \mathbb{Z}_+$  and edge costs  $w_e \in \mathbb{Z}_+$  for all  $e \in E$ , and let  $K \subseteq V^{(2)}$  be a set of undirected commodities with demand values  $d_{s,t} \in \mathbb{Z}_+$  for all  $(s, t) \in K$ . Consider

the following linear programming relaxation of (the multicommodity flow relaxation of) the undirected CAP-USPR problem:

$$\begin{aligned}
 \min \quad & \sum_{e \in E} w_e z_e & (\text{CAP-LP}) \\
 \sum_{P \in \mathcal{P}(s,t)} x_P &= 1 & \forall (s,t) \in K \\
 \sum_{(s,t) \in K} \sum_{P \in \mathcal{P}(s,t): e \in P} d_{s,t} \cdot x_P &\leq u_e z_e & \forall e \in E \\
 z_e &\geq 0 & \forall e \in E \\
 0 \leq x_P &\leq 1 & \forall (s,t) \in K, P \in \mathcal{P}(s,t)
 \end{aligned} \tag{6.5}$$

The idea of our approximation algorithm for CAP-USPR is to round an optimal solution  $(x^*, z^*)$  of (CAP-LP) to an integer solution  $([z], [x])$  of (CAP-LP) in such a way that the corresponding routing paths  $S := \{P : [x]_P = 1\}$  form a USPR.

It is well known (and follows directly from LP duality) that an optimal solution of (CAP-LP) is given by routing each commodity  $(s, t)$  on shortest path  $P_{(s,t)}^*$  with respect to the edge lengths  $w_e/u_e$  and installing  $z_e := \sum_{(s,t): e \in P_{(s,t)}^*} d_{s,t}/u$  many capacity units on each edge  $e \in E$ . For an appropriate perturbation of these edge lengths, all shortest paths are unique and form a USPR. In the corresponding optimal solution  $(x^*, z^*)$  of (CAP-LP), all  $x_P$  variables are integer. Only the  $z_e$  variables may attain fractional values and need to be rounded up.

Unfortunately, the optimal fractional solution values  $z_e^*$  may be arbitrarily small, so rounding them up may increase the total cost by an arbitrarily large factor. Yet, if we knew which edges are contained in an optimal CAP-USPR solution  $(z^{\text{opt}}, x^{\text{opt}})$  in advance, then we could easily restrict the linear program (CAP-LP) to allow capacity installation on only these edges. The additional cost of rounding up the capacity multipliers  $z_e^*$  of the optimal linear programming solution then would be bounded by the cost of the optimal solution, and we would obtain a 2-approximate solution, in the worst case.

In the special case where the underlying graph is an undirected cycle, we do not need to know the optimal solution's topology in advance. It either contains all edges of the cycle or it is contained in one of the paths obtained by removing one of the edges of the cycle. In algorithm `EnumerateAndRound` shown on the next page, we simply enumerate all these possibilities.

**Theorem 6.21** *Algorithm `EnumerateAndRound` is a 2-approximation algorithm for CAP-USPR on an undirected cycle.*

**Proof.** For each  $l \in E$ , the metric  $\lambda^l$  defined in Step 2.1 of the algorithm clearly induces a USPR whose paths do not contain  $l$ . The metric  $\lambda^0$  defined

in Step 1.1 is a perturbation of the metric  $w_e/u_e$ . Since  $\lambda_{e_0}^0$  is odd and all other lengths  $\lambda_e^0$  are even, also  $\lambda^0$  defines unique shortest paths between all node pairs. Hence, all metrics  $\lambda^0$  and  $\lambda^l$  with  $l \in E$  define valid USPRs and the corresponding solutions  $(\lambda^0, z^0)$  and  $(\lambda^l, z^l)$  are feasible.

Let  $(\lambda^{opt}, z^{opt})$  be the optimal solution of the given CAP-USPR instance. We consider two possible cases.

First, assume that  $z_e^{opt} \geq 1$  for all  $e \in E$ . In this case, we have

$$\begin{aligned} w(z^0) &= \sum_{e \in E} w_e z_e^0 = \sum_{e \in E} w_e \lceil f_e(\lambda^0)/u_e \rceil \\ &\leq \sum_{e \in E} w_e (f_e(\lambda^0)/u_e + z_e^{opt}) = w(z^*) + w(z^{opt}) \leq 2w(z^{opt}), \end{aligned}$$

which implies that  $(\lambda^0, z^0)$  is a 2-approximate solution.

If this is not the case, we have  $z_l^{opt} = 0$  for some  $l \in E$ . As  $G-l$  is a path, the routing of all commodities is uniquely determined in this case. Hence, the metric  $\lambda^l$  constructed in algorithm **EnumerateAndRound** and the optimal solution's metric  $\lambda^{opt}$  induce the same shortest paths and, thus, define the same solution. As  $z^l$  is a minimum cost capacity installation for this routing,  $(\lambda^l, z^l)$  even is an optimal solution in this case.  $\square$

Again, Theorem 6.21 generalizes straightforward to undirected CAP-USPR instances where all blocks of the underlying graph are cycles.

---

**Algorithm 6.3** EnumerateAndRound

---

**Input:** Undirected cycle  $G = (V, E)$  with capacities  $u_e \in \mathbb{Z}_+$  and costs  $w_e \in \mathbb{Z}_+$  for all  $e \in E$  and an undirected commodity set  $K \subseteq V^{(2)}$  with demands  $d_{s,t} \in \mathbb{Z}_+$  for all  $(s, t) \in K$ .

**Output:** Capacity multipliers  $z_e \in \mathbb{Z}_+$  and a valid metric  $\lambda \in \mathbb{Z}_+^E$  for the commodity set  $K$ , such that the induced edge flows  $f_e(\lambda)$  do not exceed the capacities  $u_e z_e$  for all  $e \in E$ .

1. Compute solution  $(\lambda^0, z^0)$  as follows:
    - 1.1 Set  $\lambda_e^0 := \begin{cases} Mw_e/u_e + 1, & \text{if } e = e_0, \\ Mw_e/u_e, & \text{otherwise,} \end{cases}$   
for some arbitrary  $e_0 \in E$  and  $M := 2 \prod_{e \in E} u_e$ .
    - 1.2 Set  $z_e^0 := \lceil f_e(\lambda^0)/u_e \rceil$  for all  $e \in E$ .
  2. For each  $l \in E$ , compute solution  $(\lambda^l, z^l)$  as follows:
    - 2.1 Set  $\lambda_e^l := \begin{cases} |E|, & \text{if } e = l, \\ 1, & \text{otherwise.} \end{cases}$
    - 2.2 Set  $z_e^l := \lceil f_e(\lambda^l)/u_e \rceil$  for all  $e \in E$ .
  3. Return minimum cost solution of  $(\lambda^0, z^0)$  and  $(\lambda^l, z^l)$ ,  $l \in E$ .
-

### 6.6.3 Min-Con-USPR on a Bidirected Ring

For MIN-CON-USPR on a bidirected ring, neither the perturbation technique used in algorithm `PenalizeSmallLinks` nor the naive rounding of an optimal solution of the linear programming relaxation alone lead to a constant factor approximation: The perturbation technique produces  $|A|/2$ -approximate solutions in the worst case, and the rounding approach may produce non-USPR integer routings as illustrated in Example 6.5.1.

The idea of our algorithm is to remove some arc from the given bidirected ring and round the optimal fractional routing in the residual digraph. The following two lemmas show that any routing obtained this way is a valid USPR. Recall that two paths  $P_1, P_2 \in \mathcal{P}$  are said to be *conflicting*, if there are two nodes  $u, v \in V$  such that both  $P_1$  and  $P_2$  contain an  $(u, v)$ -subpath  $P_1[u, v]$  and  $P_2[u, v]$ , respectively, but  $P_1[u, v] \neq P_2[u, v]$ .

**Lemma 6.22** *Let  $D = (V, A)$  be a bidirected ring. Then there exists an optimal solution  $(x^*, L^*)$  of (CON-LP) such that  $x_{P_1}^* = 0$  or  $x_{P_2}^* = 0$  for any pair of conflicting paths  $P_1$  and  $P_2$  with  $(s_{P_1}, t_{P_1}) \neq (s_{P_2}, t_{P_2})$ . Furthermore, such a solution  $(x^*, L^*)$  can be found in polynomial time.*

**Proof.** Suppose we have an optimal solution  $(x^*, L^*)$  of (CON-LP) with  $x_{P_1}^* > 0$  and  $x_{P_2}^* > 0$  for two conflicting paths  $P_1$  and  $P_2$  with  $(s_{P_1}, t_{P_1}) \neq (s_{P_2}, t_{P_2})$ . For simplicity, we denote  $s_1 := s_{P_1}$ ,  $t_1 := t_{P_1}$ ,  $s_2 := s_{P_2}$ , and  $t_2 := t_{P_2}$ .

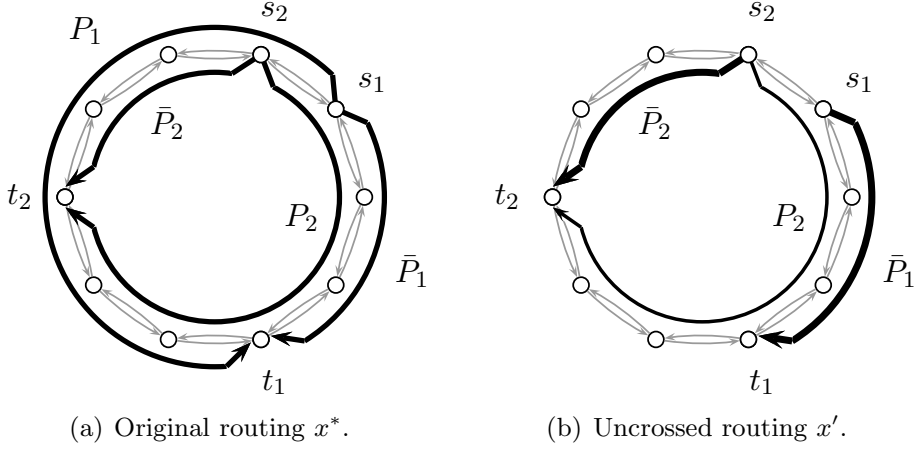
Let  $\bar{P}_1$  be the opposite  $(s_1, t_1)$ -path to  $P_1$  and let  $\bar{P}_2$  be the opposite  $(s_2, t_2)$ -path to  $P_2$ . Since  $P_1$  and  $P_2$  conflict, we have  $\bar{P}_1 \subsetneq P_2$  and  $\bar{P}_2 \subsetneq P_1$ , see Figure 6.14. We may assume w.l.o.g. that  $P_1$  carries less flow than  $P_2$  in the routing given by  $x^*$ , i.e.,  $d_{s_1, t_1} x_{P_1}^* \leq d_{s_2, t_2} x_{P_2}^*$ . Let  $\alpha := d_{s_1, t_1} x_{P_1}^*$ .

We now construct another optimal solution  $(x', L^*)$  of (CON-LP) by ‘uncrossing’ the routing of the two commodities  $(s_1, t_1)$  and  $(s_2, t_2)$ , as shown in Figure 6.14. For commodity  $(s_1, t_1)$ , we shift the entire flow of value  $\alpha$  from path  $P_1$  to its opposite path  $\bar{P}_1$ . Simultaneously, we also shift a flow of value  $\alpha$  from  $P_2$  to  $\bar{P}_2$  for commodity  $(s_2, t_2)$ . Formally,  $x'$  is given as

$$\begin{aligned} x'_{P_1} &:= 0, & x'_{P_2} &:= x_{P_2}^* + \alpha/d_{s_2, t_2}, \\ x'_{\bar{P}_1} &:= 1, & x'_{\bar{P}_2} &:= x_{\bar{P}_2}^* - \alpha/d_{s_2, t_2}, \text{ and} \\ x'_P &:= x_P^*, & & \text{for all } P \notin \{P_1, \bar{P}_1, P_2, \bar{P}_2\}. \end{aligned}$$

One easily verifies that  $f_a(x') \leq f_a(x^*)$  for all  $a \in A$ , so  $x'$  is an optimal solution of (CON-LP), too. Note that this uncrossing operation reduces the total flow in the network, i.e.,  $\sum_{a \in A} f_a(x') < \sum_{a \in A} f_a(x^*)$ .

Thus, in any optimal solution  $x^*$  of (CON-LP) which in addition also minimizes  $\sum_{a \in A} f_a(x)$  (over all optimal solutions of (CON-LP)) we have  $x_{P_1}^* = 0$  or  $x_{P_2}^* = 0$  for any pair of conflicting paths  $P_1$  and  $P_2$  with  $(s_{P_1}, t_{P_1}) \neq (s_{P_2}, t_{P_2})$ .



**Figure 6.14** Uncrossing the routing of two parallel commodities.

We can easily find such a solution  $x^*$  as follows: First, we solve the linear program (CON-LP) to determine the optimal value  $L^*$ . Then, we solve (CON-LP) with an additional linear constraint  $L \leq L^*$  and the objective function replaced by  $\min \sum_{a \in A} f_a(x)$ . The optimal solution  $x^*$  of the second linear program then has the required properties.  $\square$

**Lemma 6.23** *Let  $D = (V, A)$  be a bidirected ring and  $a_0 \in A$ . Let  $S$  be a set of paths in  $D - a_0$  that contains no pair of conflicting paths. Then there exists a compatible metric  $\lambda \in \mathbb{Z}_+^A$  for  $S$ .*

**Proof.** Without loss of generality, we may assume that the nodes of  $D$  are labeled  $v_1$  to  $v_n$  in clockwise order and that  $a_0 = (v_n, v_1)$ . The anti-parallel arc of  $a_0$  is denoted  $a_1 = (v_1, v_n)$ .

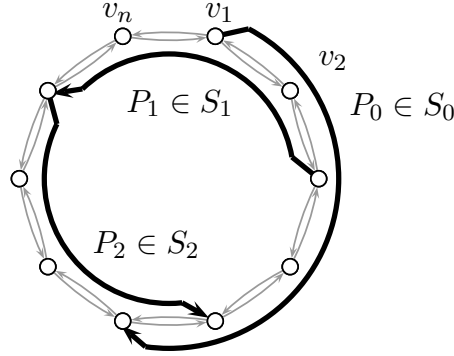
Suppose  $a_1 \notin P$  for all  $P \in S$ . Then all paths  $P \in S$  are unique shortest paths for the metric

$$\lambda_a := \begin{cases} |V|, & \text{if } a \in \{a_0, a_1\}, \text{ and} \\ 1, & \text{otherwise.} \end{cases}$$

So, we may assume that  $a_1 \in P$  for some  $P \in S$ . We distinguish three types of paths in  $S$ :

$$\begin{aligned} S_0 &:= \{P \in S : P = (v_i, v_{i+1}, \dots, v_j) \text{ with } i < j\}, \\ S_1 &:= \{P \in S : a_1 \in P\}, \text{ and} \\ S_2 &:= \{P \in S : P = (v_j, v_{j-1}, \dots, v_i) \text{ with } i < j\}. \end{aligned}$$

The set  $S_0$  consists of all clockwise oriented paths of  $S$ , the set  $S_1$  of all counter-clockwise oriented paths of  $S$  that contain  $a_1$ , and the set  $S_2$  of all remaining counter-clockwise oriented paths of  $S$ , see Figure 6.15. As no path of  $S$  contains  $a_0$ , the three sets  $S_0$ ,  $S_1$ , and  $S_2$  form a partition of  $S$ .



**Figure 6.15** Path sets  $S_0$ ,  $S_1$ , and  $S_2$  in a bidirected ring.

First, we show that there is a compatible metric for the path set  $S_0 \cup S_1$ . For this, we consider these paths in an undirected setting. Let  $G = (V, E)$  be the undirected cycle with  $E := \{v_i v_{i+1} : i = 1, \dots, n\}$ , where  $v_{n+1} = v_1$ . Recall that  $a_1 \in P$  for all  $P \in S_1$ , and  $a_1 \notin P$  for all  $P \in S_0$ . Hence, there are no two directed paths in  $S_0 \cup S_1$  that correspond to the same undirected path. Furthermore, the set of undirected paths corresponding to  $S_0 \cup S_1$  contains no pair of conflicting undirected paths. It was shown by Ben-Ameur and Gourdin [27] that in this case there exists a metric  $\lambda' \in \mathbb{R}_+^E$  in the undirected cycle  $G$  that is compatible with the undirected path set corresponding to  $S_0 \cup S_1$ , see Theorem 5.26 in Section 5. Its corresponding directed metric  $\lambda'' \in \mathbb{R}_+^A$  given as  $\lambda''_{(u,v)} := \lambda'_{uv}$  for all  $(u, v) \in A$  is compatible with the directed path set  $S_0 \cup S_1$  in  $D$ .

Now, we modify this metric such that it is compatible with the entire path set  $S = S_0 \cup S_1 \cup S_2$ . Recall that no path in  $S_0 \cup S_1$  contains the arc  $a_0$ . Thus, also the metric  $\lambda \in \mathbb{R}_+^A$  defined as

$$\lambda_a := \begin{cases} \lambda''_a, & \text{if } a \neq a_0, \text{ and} \\ \sum_{a \in A \setminus \{a_0\}} \lambda''_a + 1, & \text{if } a = a_0, \end{cases}$$

is compatible with  $S_0 \cup S_1$ . Furthermore, note that, for each path  $P \in S_2$ , the corresponding clockwise oriented  $(s_P, t_P)$ -path  $\bar{P}$  contains the arc  $a_0$ . Hence, each path  $P \in S_2$  is shorter than its counterpart  $\bar{P}$  with respect to  $\lambda$ , which implies that  $\lambda$  is compatible with entire path set  $S$ .  $\square$

Lemma 6.22 and Lemma 6.23 lead to the constant factor approximation algorithm **BidirectedRingRounding** shown on the following page. In the first step of this algorithm, we compute an optimal multicommodity flow routing  $x^*$  with the additional properties stated in Lemma 6.22. Then we remove the least utilized arc  $a_{\min} = \arg \min f_a(x^*)$  from the bidirected ring and 'round' the optimal MCF routing in such a way, that no routing path uses the arc  $a_{\min}$  and that no pairs of conflicting paths are created. By Lemma 6.23, the routing obtained this way is indeed a USPR for the given commodities. Using the methods presented in Chapter 4, we then compute a compatible

metric for this routing.

**Theorem 6.24** *BidirectedRingRounding is a 3-approximation algorithm for MIN-CON-USPR on a bidirected ring.*

**Proof.** It follows immediately from Lemmas 6.22 and 6.23 that algorithm BidirectedRingRounding computes a valid solution for MIN-CON-USPR. It remains to show that this solution has a congestion of at most three times the optimal solution's congestion.

In Step 3 of algorithm BidirectedRingRounding, we shift all flows on paths across arc  $a_{\min}$  to the respective opposite flow paths, and we round path variables  $x_p$  with  $x_p^* \geq 0.5$  to 1. Hence, for any arc  $a \in A$ , we have  $f_a([x]) \leq 2f_a(x^*) + f_{a_{\min}}(x^*) \leq 3f_a(x^*)$ .  $\square$

Algorithm BidirectedRingRounding and Theorem 6.24 straightforward carry over to the case where all strongly connected components of  $D$  are bidirected rings or subgraphs of bidirected rings.

#### 6.6.4 Cap-USPR on a Bidirected Ring

In this final section, we show how to approximate CAP-USPR on a bidirected ring within a constant factor. In principle, we use the same approach as for the undirected problem version: We compute a solution  $(\lambda^0, z^0)$  that is a 2-approximation of the optimal solution  $(\lambda^{opt}, z^{opt})$  in the case where

---

##### Algorithm 6.4 BidirectedRingRounding

---

**Input:** Bidirected ring  $D = (V, A)$  with capacities  $u_a \in \mathbb{Z}_+$  for all arcs  $a \in A$  and a commodity set  $K \subseteq V \times V$  with demands  $d_{s,t} \in \mathbb{Z}_+$  for all  $(s, t) \in K$ .

**Output:** A valid metric  $\lambda \in \mathbb{Z}_+^A$  for the commodity set  $K$ .

1. Compute an optimal solution  $(x^*, L^*)$  of (CON-LP) with  $x_{P_1}^* = 0$  or  $x_{P_2}^* = 0$  for any pair of conflicting paths  $P_1 \in \mathcal{P}(s_1, t_1)$  and  $P_2 \in \mathcal{P}(s_2, t_2)$  with  $(s_1, t_1) \neq (s_2, t_2)$ .
2. Find arc  $a_{\min} := \arg \min f_a(x^*)$ .  
Let  $\bar{a}_{\min}$  be the reverse arc of  $a_{\min}$ .
3. Define  $[x] \in \{0, 1\}^{\mathcal{P}(K)}$  as

$$[x]_P := \begin{cases} 0, & \text{if } a_{\min} \in P \text{ or} \\ & \text{if } x_P^* < 0.5 \text{ or} \\ & \text{if } x_P^* = 0.5 \text{ and } \bar{a}_{\min} \in P, \text{ and} \\ 1, & \text{otherwise.} \end{cases}$$

4. Find a compatible metric  $\lambda$  for  $S := \{P \in \mathcal{P}(K) : [x]_P = 1\}$ .
  5. Return  $\lambda$ .
-



$z_a^{opt} \geq 1$  for all  $a \in A$ , and  $|A|$  many solutions  $(\lambda^l, z^l)$ ,  $l \in A$ , to cope with the cases where  $z_l^{opt} = 0$  for some  $l \in A$ . In contrast to the CAP-USPR problem on an undirected cycle, however, the CAP-USPR problem on a bidirected ring remains  $\mathcal{NP}$ -hard even in the restricted case with  $z_l = 0$  for some arc  $l \in A$ . Nevertheless, it is possible to approximate the problem in this restricted case within a constant factor by rounding the optimal solution of the following linear programming relaxation of CAP-USPR:

$$\begin{aligned}
& \min \sum_{a \in A} w_a z_a && \text{(CAP-LP2)} \\
& \sum_{P \in \mathcal{P}(s,t)} x_P = 1 && \forall (s,t) \in K \\
& \sum_{(s,t) \in K} \sum_{P \in \mathcal{P}(s,t): a \in P} d_{s,t} \cdot x_P \leq u_a z_a && \forall a \in A \\
& x_P \leq z_a && \forall P \in \mathcal{P}(K), a \in P \\
& z_a \geq 0 && \forall a \in A \\
& 0 \leq x_P \leq 1 && \forall (s,t) \in K, P \in \mathcal{P}(s,t)
\end{aligned} \tag{6.6}$$

The linear program (CAP-LP2) is the directed version of (CAP-LP) with some additional inequalities of type (6.6). These inequalities are trivially valid for any integer solution of (the directed version of) (CAP-LP). So, any optimal solution of (CAP-LP2) provides a lower bound on the optimal solution value of a given CAP-USPR instance. If  $D$  is a bidirected ring, these inequalities close a substantial part of the integrality gap of (CAP-LP), while the strengthened formulation remains polynomially solvable.

Analogous to Lemma 6.22, one can prove the existence of an optimal solution  $(x^*, z^*)$  of (CAP-LP) (without inequalities (6.6)) with  $x_{P_1}^* = 0$  or  $x_{P_2}^* = 0$  for any pair of conflicting paths  $P_1$  and  $P_2$  with  $(s_{P_1}, t_{P_1}) \neq (s_{P_2}, t_{P_2})$ . For the stronger formulation (CAP-LP2), one obtains the following weaker result.

**Lemma 6.25** *Let  $D = (V, A)$  be a bidirected ring. Then there exists an optimal solution  $(x^*, z^*)$  of (CAP-LP2) such that  $x_{P_1}^* + x_{P_2}^* \leq 1$  for any pair of conflicting paths  $P_1$  and  $P_2$  with  $(s_{P_1}, t_{P_1}) \neq (s_{P_2}, t_{P_2})$ . Furthermore, such a solution  $(x^*, z^*)$  can be found in polynomial time.*

**Proof.** Analogous to the proof of Lemma 6.22. □

Just as in the previous section, any optimal solution of (CAP-LP2) with the properties stated in Lemma 6.25 can be turned into a valid solution of the given CAP-USPR problem instance by removing one arc, rounding the fractional flows in the remaining digraph consistently, and finally installing arc capacities that are sufficient for the resulting flows. The corresponding algorithm `BidirectedEnumerateAndRound` is illustrated on the next page.

**Theorem 6.26** *BidirectedEnumerateAndRound is a 4-approximation algorithm for CAP-USPR on a bidirected ring.*

**Proof.** The proof of Theorem 6.26 is similar to that of Theorem 6.24.

It follows from Lemma 6.23 and Lemma 6.25 that, for each  $l \in A$ , the path set  $S^l$  computed in Step 2.4 defines a USPR for the given commodity set  $K$  and that  $S^l$  is a compatible metric for this routing.

The metric  $\lambda^0$  defined in Step 1.2 is a perturbation of the metric  $w_a/u_a$ . As  $\lambda_{(v_0, v_1)}^0$  and  $\lambda_{(v_1, v_0)}^0$  are odd and all other lengths  $\lambda_a^0$  are even, also  $\lambda^0$  defines a USPR. Hence, all solutions  $(\lambda^0, z^0)$  and  $(\lambda^l, z^l)$  are feasible.

Now consider the optimal solution  $(\lambda^{opt}, z^{opt})$  of the given CAP-USPR instance. Analogous to the undirected case, it follows that  $(\lambda^0, z^0)$  is a

---

**Algorithm 6.5** BidirectedEnumerateAndRound

---

**Input:** Bidirected ring  $D = (V, A)$  with capacities  $u_a \in \mathbb{Z}_+$  and costs  $w_a \in \mathbb{Z}_+$  for all  $a \in A$  and a commodity set  $K \subseteq V \times V$  with demands  $d_{s,t} \in \mathbb{Z}_+$  for all  $(s, t) \in K$ .

**Output:** Capacity multipliers  $z_a \in \mathbb{Z}_+$  and a valid metric  $\lambda \in \mathbb{Z}_+^A$  for the commodity set  $K$ , such that the induced arc flows  $f_a(\lambda)$  do not exceed the capacities  $u_a z_a$  for all  $a \in A$ .

1. Compute solution  $(\lambda^0, z^0)$  as follows:

1.1 Set  $\lambda_a^0 := \begin{cases} Mw_a/u_a + 1, & \text{if } a \in \{(v_0, v_1), (v_1, v_0)\}, \\ Mw_a/u_a, & \text{otherwise,} \end{cases}$   
with  $M := 2\prod_{a \in A} u_a$ .

1.2 Set  $z_a^0 := \lceil f_a(\lambda^0)/u_a \rceil$  for all  $a \in A$ .

2. For each  $l \in A$ , compute solution  $(\lambda^l, z^l)$  as follows:

2.1 Compute an optimal solution  $(z^{l*}, x^{l*})$  of (CAP-LP2) with the restriction  $z_l^l = 0$  and such that  $x_{P_1}^{l*} + x_{P_2}^{l*} \leq 1$  for any pair of conflicting paths  $P_1 \in \mathcal{P}(s_1, t_1)$  and  $P_2 \in \mathcal{P}(s_2, t_2)$  with  $(s_1, t_1) \neq (s_2, t_2)$ .

2.2 Let  $\bar{l}$  be the reverse arc of  $l$ .

2.3 Define  $[x^l] \in \{0, 1\}^{\mathcal{P}(K)}$  as

$$[x]_P^l := \begin{cases} 0, & \text{if } l \in P \text{ or} \\ & \text{if } x_P^{l*} < 0.5 \text{ or} \\ & \text{if } x_P^{l*} = 0.5 \text{ and } \bar{l} \in P, \text{ and} \\ 1, & \text{otherwise.} \end{cases}$$

2.4. Find a compatible metric  $\lambda^l$  for  $S^l := \{P \in \mathcal{P}(K) : [x]_P^l = 1\}$ .

2.5 Set  $z_a^l := \lceil f_a(\lambda^l)/u_a \rceil$  for all  $a \in A$ .

3. Return minimum cost solution of  $(\lambda^0, z^0)$  and  $(\lambda^l, z^l)$ ,  $l \in A$ .

---

2-approximate solution if  $z_a^{opt} \geq 1$  for all  $a \in A$ .

So, assume that  $z_l^{opt} = 0$  for some  $l \in A$ . For the corresponding solution  $(z^{l*}, x^{l*})$  of (CAP-LP2), the inequalities (6.6) then imply that  $z_a^{l*} \geq 0.5$  for each arc  $a \in A$  with  $\lceil z_a^{l*} \rceil \geq 1$ . Hence, we have

$$w(z^l) = \sum_{a \in A} w_a z_a^l = \sum_{a \in A} w_a \lceil z_a^{l*} \rceil \leq \sum_{a \in A} w_a 4z_a^{l*} = 4w(z^{l*}) \leq 4w(z^{opt}) ,$$

which implies that  $(\lambda^l, z^l)$  is a 4-approximate solution in this case.  $\square$

Again, algorithm `BidirectedEnumerateAndRound` and Theorem 6.26 generalize to digraphs in which all strongly connected components are bidirected rings.

## 6.7 Concluding Remarks

In this chapter, we have shown that it is  $\mathcal{NP}$ -hard to approximate MIN-CON-USPR within a factor of  $\mathcal{O}(|V|^{1-\epsilon})$  in general and CAP-USPR within a factor of  $\mathcal{O}(2^{\log^{1-\epsilon} |V|})$  in the directed or  $2 - \epsilon$  in the undirected case. The fixed charge network design problem FC-USPR was proven to be  $\mathcal{NPO}$ -complete. We presented simple  $|A|$ - and  $|K|$ -approximation algorithms for MIN-CON-USPR in general networks and we illustrated how known techniques can be used to approximate several restricted versions of CAP-USPR. For the special cases where the underlying graph is an undirected cycle or a bidirected ring, constant factor approximation algorithms for MIN-CON-USPR and CAP-USPR were proposed.

We also constructed examples where the minimum congestion obtainable with unsplittable shortest path routing is a factor of  $\Omega(|V|^2)$  larger than the congestion of an optimal unsplittable flow routing or an optimal shortest multi-path routing, and a factor of  $\Omega(|V|)$  larger than the congestion of an optimal unsplittable source-invariant routing.

It remains open whether the inapproximability threshold of  $\Omega(|V|^{1-\epsilon})$  for MIN-CON-USPR is tight or whether better approximation ratios than  $\mathcal{O}(\min\{|A|, |K|\})$  can be achieved. It is also not known how to compute approximate solutions with reasonable quality guarantees for the general CAP-USPR problem. The methods known for the corresponding GENERALIZED STEINER NETWORK or BUY-AT-BULK NETWORK DESIGN problem versions do not necessarily yield feasible solutions for the CAP-USPR problem.



## Part III

# An Integer Programming Solution Approach



## Chapter 7

# Integer Linear Programming Models

In this chapter, we introduce several integer linear programming formulations for unsplittable shortest path routing problems, study the algorithmic properties of these formulations, and compare the strength of their respective linear relaxations.

We present two basic types of integer linear programming formulations. The first class of formulations uses binary path variables to describe the set of valid unsplittable shortest path routings, whereas the second class uses binary arc routing variables. We also discuss a variant of the second class where additional binary variables are used to describe the forwarding that is induced by the routing (i.e., the arcs via which the shortest paths leave each intermediate node). For each variant, we discuss the formulation of unsplittable shortest path routings in general and present corresponding (mixed-) integer linear programming formulations for the three basic unsplittable shortest path routing problems FC-USPR, CAP-USPR, and MIN-CON-USPR, that have been introduced in Chapter 3 of this thesis. Formulations for other unsplittable shortest path routing problems can be derived analogously.

In contrast to the ‘traditional’ integer linear programming models for shortest path routing problems, our formulations contain no variables for the routing lengths. Instead, our formulations rely on special linear inequalities to ensure that the paths described by a feasible integer solution indeed form an unsplittable shortest path routing. This approach leads to integer linear programs that are not only much smaller but also stronger than those obtained with the traditional formulations. Once our models are solved and the paths of an unsplittable shortest path routing are found, a compatible routing metric can be easily computed in a post-processing step, using the methods presented in Chapter 4 of this thesis. Virtually, 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 valid metric that induces exactly these paths.

The remainder of this chapter is organized as follows. In Section 7.1, we review the existing literature on integer linear programming models for unsplittable and multi-shortest path routing problem. Section 7.2 contains a brief review the basic necessary notation and illustrates the general modeling and solution approach proposed.

In Section 7.3, we introduce our first class of integer linear programming formulations, which are based on binary path routing variables and the independence system characterization of the path sets of unique shortest path routings. The second class of formulations, in which the valid unsplittable shortest path routings are described by binary arc routing variables, is presented in Section 7.4.

In Section 7.5 we discuss the computational complexity of the separation and pricing associated with these formulations. We show that in both formulations the separation problem associated with those model inequalities that ensure that the paths form an unsplittable shortest path routing is  $\mathcal{NP}$ -hard in general, but solvable in polynomial time if the point is integer. We also show that the pricing problem for the path variables in the first class of formulations can be solved in polynomial time with respect to the size of the current restricted linear relaxation, but that an optimal solution of the entire linear relaxation may involve exponentially many active path variables and exponentially many binding linear inequalities. Assuming that  $\mathcal{P} \neq \mathcal{NP}$ , we thus may not even expect the linear programming relaxations arising in a branch-and-bound algorithm based on these formulations to be solvable in polynomial time.

In Section 7.6, we finally compare the strength of the proposed formulations. We show that the two variants of the arc routing formulations are equivalent with respect to their linear relaxation values, whereas the path routing and the arc routing formulations are incomparable in general.

## 7.1 Related Work

Integer and mixed integer linear programming nowadays is one of the standard tools for solving network design and routing problems in telecommunications. There exists a vast literature related to integer linear programming models and methods for various kinds of network design and routing problems. The scope ranges from general network flow and connectivity problems, such as minimum cost flow or Steiner network problems, to very specialized problems addressing issues related to network survivability or particular hardware and routing technologies at a very detailed level. An overview of the fundamental methodology and some recent results can be



found in the textbooks and surveys of Grötschel et al. [105], Pióro and Medhi [162] or Resende and Pardalos [172], for example.

Shortest path routing problems have been rarely tackled with integer programming methods until recently. Instead, local search algorithms or other meta-heuristics and Lagrangian relaxation techniques have been very popular to solve these problems [26, 34, 37, 48, 81, 92, 93, 133, 163]. Fortz and Thorup, Fortz and Thorup, who considered a traffic engineering problem with multi-shortest path routing in [92, 93], also computed lower bounds by solving the corresponding fractional multicommodity flow relaxation.

The first integer programming formulation of a shortest path routing problem (to our knowledge) was presented by Bley et al. [37], who considered a (survivable) network design problem with unsplittable shortest path routing. This formulation contains binary arc routing variables for the commodities' end-to-end routing paths, integer arc length variables for the routing metric, and continuous variables for the distances between the node pairs with respect to these routing lengths. With these variables, the interdependencies between the routing path variables and the arc length and distance variables can be easily expressed by quadratic constraints. These have been linearized in a straightforward manner, leading to a quadratic number of linear constraints involving big-M coefficients. The uniqueness of the shortest paths was guaranteed by adding a pre-determined perturbation to the integer arc length variables.

De Giovanni et al. [69] used a similar formulation to solve a minimum cost network design problem for the OSPF routing variant with equal cost multi-path (ECMP) traffic splitting, as described in Chapter 3 on page 27. The commodities in this model are given by the traffic destination nodes, i.e., all traffic demands with a common destination are aggregated into one multi-source single-destination commodity. In the routing is expressed by binary variables indicating which arcs are contained in the routing paths towards each destination, continuous variables for each commodities actual arc flows, and continuous variables for the routing lengths of the arcs and for the shortest path distances with respect to these lengths. In this model, both the relation between the binary arc routing variables and the continuous arc flow variables as well as the relation between the binary arc routing variables and the continuous routing length and distance variables are expressed via big-M (in)equalities. As this multi-path routing variant does not require the uniqueness of the shortest paths, the formulation of de Giovanni et al. does not contain any constraints related to this particularity of unsplittable shortest path routing. Instead, it contains a set of additional traffic splitting constraints, which ensure that each commodity's flow is split evenly among the outgoing routing arcs at each intermediate node. Based on an analogous formulation, Parmar et al. [158] recently proposed an integer programming approach for the problem of finding a multi-shortest path routing with even traffic splitting that minimized the congestion in a

network with given capacities.

Similar formulations were also presented by Bourquia et al. [41], Pióro et al. [163], and Tomaszewski et al. [184] for the problem of finding an unsplittable shortest path routing that minimizes the maximum congestion and by Eremin et al. [80] for the problem of finding a  $k$ -splittable such routing.

From an algorithmic point of view, these ‘traditional’ integer linear programming formulations have several severe disadvantages. The (implicit) step of linearizing the quadratic constraints, which express the relation between the routing paths and the routing lengths, leads to huge integer linear formulations, even for very small networks. Furthermore, the resulting linear constraints employ big-M coefficients to link the binary arc or path routing variables to the continuous or general integer routing length and distance variables. With respect to their linear relaxation values, these formulations are very weak. Standard branch-and-bound based integer linear programming approaches usually perform very poorly on such formulations. In [37], Bley et al. used this type of integer programming formulation only to formally describe the considered (survivable) network design problem. For solving it, they retreated to a local search heuristic. de Giovanni et al. [69], Bourquia et al. [41], and Eremin et al. [80] each solved the proposed models using standard integer programming techniques. Their computational results reveal that these models are not well-suited for solving instances of practical size, unless they are substantially strengthened using additional inequalities.

A first attempt to formulate unsplittable shortest path routing problems as an integer linear program without variables for the routing metric was made by Staehle et al. [180] and Milbrandt [141], who proposed integer linear programming models with only binary arc-flow variables for the problem of finding an unsplittable shortest path routing that minimizes a linear combination of the maximum and the average link congestion. These models ensure that the computed routing paths satisfy the Bellman property, but they still admit non-unsplittable shortest path routings as integer solution.

The first correct integer linear programming approach without additional variables for the routing metric was presented by Bley and Koch [33, 36] for a survivable network design problem with unsplittable shortest path routing. Besides the link capacity variables, their model contains only binary arc routing variables for each commodity. In order to ensure that a valid unsplittable shortest path routings is obtained, they separate inequalities that cut-off invalid routings. The arc routing formulation presented in Section 7.4 below is based on this model. Holmberg and Yuan [115] generalized this approach to solve a network design problem with unsplittable shortest path routing and multi-cast commodities.

In addition, there exist a large number of publications that address unsplittable shortest path routing problems in which all commodities share a

common source and, in addition, the chosen routing paths must to form a tree. Problems of this type arise in the planning of access and tributary networks for telecommunications and electric energy, for example, and are typically referred to as access network design or hub location problems in this context. They can be regarded as single-source variants of unsplittable shortest path routing problems. Here, we just want to mention the publications of Kleinberg [123], Dinitz et al. [74] and Skutella [179], who mainly address the approximability of such unsplittable flow problems, and the works of van de Leensel [186], Pfender [160], and Labbé and Yaman [129, 128, 130], who discuss integer programming formulations, polyhedral results, and dynamic programming solution approaches for these problems.

## 7.2 Basics

Before we turn our attention to our general integer programming approach for the fixed charge network design problem FC-USPR, let us briefly review the basic notions, notations, and concepts introduced in the preceding chapters.

Let  $D = (V, A)$  be a directed graph with non-negative arc capacities  $u = (u_a)_{a \in A} \in \mathbb{Z}_+^A$  and arc costs  $w = (w_a)_{a \in A} \in \mathbb{Z}_+^A$ . Furthermore, let  $K \subseteq V \times V$  be a set of directed commodities with demand values  $d_{s,t} \in \mathbb{Q}_+$ ,  $(s, t) \in K$ . Without loss of generality, we may assume that all arc capacities  $u_a$ ,  $a \in A$ , and all demand values  $d_{s,t}$ ,  $(s, t) \in K$ , are strictly positive. For each commodity  $(s, t) \in K$ , let  $\mathcal{P}(s, t)$  be the set of all simple  $(s, t)$ -paths in  $D$ . We let  $\mathcal{P}(K) := \bigcup_{(s,t) \in K} \mathcal{P}(s, t)$ . With  $\mathcal{P}$  we denote the set of all simple paths in  $D$ . Given a path  $P$ , we refer to its source node by  $s_P$  and to its target node by  $t_P$ . We write  $a \in P$  or  $v \in P$  to indicate that the arc  $a \in A$  or the node  $v \in V$  occurs in  $P$ , respectively. A vector  $\lambda \in \mathbb{R}_+^A$  is called a (routing) metric, even if it does not satisfy the triangle inequality.

A metric  $\lambda \in \mathbb{R}_+^A$  is said to be *valid* for the commodity set  $K$  if, for each commodity  $(s, t) \in K$ , the shortest  $(s, t)$ -path  $P_{(s,t)}^*(\lambda)$  with respect to  $\lambda$  is uniquely determined. Given such a valid metric, the set of these induced shortest paths forms an *unsplittable shortest path routing (USPR)* for the commodity set  $K$ , see Definition 3.4 on Page 33. The demand of each commodity  $(s, t) \in K$  is routed unsplit along the respective shortest path  $P_{(s,t)}^*(\lambda)$  in a USPR. So, given a valid metric  $\lambda$ , the total flow through an arc  $a \in A$  in its induced USPR is

$$f_a(\lambda) := \sum_{(s,t) \in K: a \in P_{(s,t)}^*(\lambda)} d_{s,t} . \quad (7.1)$$

In Chapter 3, we introduced three basic unsplittable shortest path routing problems: the congestion minimization problem MIN-CON-USPR, the capacitated network design problem CAP-USPR, and the fixed charge network design problem FC-USPR.

The task in the congestion minimization problem MIN-CON-USPR is to find a valid metric  $\lambda \in \mathbb{Z}_+^A$  for the given commodities  $K$ , such that the maximum arc congestion  $L := \max\{f_a(\lambda)/u_a : a \in A\}$  is minimized.

In the capacitated network design problem CAP-USPR, we seek for a valid metric  $\lambda \in \mathbb{Z}_+^A$  for the commodity set  $K$  and for integer capacity multipliers  $z_a \in \mathbb{Z}_+$ ,  $a \in A$ , such that  $f_a(\lambda) \leq u_a z_a$  for all  $a \in A$ . The objective is to minimize the total capacity installation cost  $\sum_{a \in A} w_a z_a$  for some given arc costs  $w_a \in \mathbb{Z}_+$ ,  $a \in A$ .

Finally, the goal in the fixed charge network design problem FC-USPR is to find a minimum cost subset of the arcs  $B \subseteq A$  and a valid metric  $\lambda \in \mathbb{Z}_+^B$ , such that (i)  $\lambda$  defines an unsplittable shortest path routing for the commodities  $K$  within the subgraph  $(V, B)$  and (ii) the induced arc flows  $f_a(\lambda)$  do not exceed the capacities  $u_a$  (on the arcs  $a \in B$ ). This clearly is equivalent to the task of finding a minimum cost vector of *binary* capacity multipliers  $z_a \in \{0, 1\}$ ,  $a \in A$ , together with a valid metric  $\lambda \in \mathbb{Z}_+^A$ , such that  $f_a(\lambda) \leq u_a z_a$  for all  $a \in A$ .

In all three problems, the solution contains a valid routing metric  $\lambda \in \mathbb{Z}_+^A$  for the given commodity set. In order to evaluate the respective objective functions, however, the metric itself is not necessary. It would be sufficient to know only the routing paths that comprise the induced USPR. (In fact, even knowing the induced arc flows would be sufficient.) Thus, an integer linear programming formulation of these problems does not necessarily have to involve a description of the routing metric. It is sufficient to model the end-to-end routing paths together with the induced arc flows and arc capacities, provided that the model guarantees that the paths of any integer feasible solution indeed comprise an unsplittable shortest path routing. Given a set of such paths, a valid metric that induces exactly these paths can be computed in polynomial time using the linear programming and rounding approach presented in Chapter 4, see Theorem 4.6 on page 47 and Theorem 4.12 on page 60. In other words, modulo the polynomial-time post-processing step of finding the actual routing metric, the three unsplittable shortest path routing problems can be formulated equivalently as a matter of finding the path set of a USPR instead of a valid metric for the given commodity set.

Accordingly, our general solution approach solves the given unsplittable shortest path routing problem in two phases: In the first phase, the end-to-end routing paths are optimized (together with the capacity installation) using integer linear programming techniques. In the second phase, the linear programming and rounding approach presented in Chapter 4 is applied to determine a valid metric that induces the paths computed in the first phase.

The models used in the first phase of this approach do not need to involve extra variables for the arc lengths of the routing metric. With the concepts introduced in Chapters 4 and 5, the path sets that comprise a USPR for the

given commodity set can be easily characterized. Recall that

- (i) a metric  $\lambda = (\lambda_a)_{a \in A} \in \mathbb{R}_+^A$  is said to be *compatible* with a given path set  $S \subseteq \mathcal{P}$ , if each path  $P \in S$  is the unique shortest  $(s_P, t_P)$ -path with respect to  $\lambda$ , and that
- (ii) a path set  $S \subseteq \mathcal{P}$  is called a *unique shortest path system (USPS)* if there exists a compatible metric  $\lambda \in \mathbb{R}_+^A$  for  $S$ .

So, a path set  $S$  is a USPR for the commodity set  $K$  if and only if it is an USPS that contains exactly one  $(s, t)$ -path for each commodity  $(s, t) \in K$ .

Alternatively, USPR can be characterized via their associated forwardings, which is useful when working with a integer programming formulation based on arc routing variables. A *forwarding* is a set  $F \subseteq V \times A$ , and each *forwarding arc*  $(t, (u, v) \in F) \in F$  means that the traffic from or via node  $u$  towards destination  $t$  must leave node  $u$  via the arc  $(u, v)$ , see Section 4.6. For a given path set  $S \subseteq \mathcal{P}$ , its *associated forwarding* is  $\mathcal{F}(S) := \bigcup_{P \in S} \{(t_P, a) : a \in P\}$ . Similar to the path set characterization,

- (i) a metric  $\lambda \in \mathbb{R}_+^A$  is said to be *compatible* with a forwarding  $F \subseteq V \times A$  if, for each  $(t, (u, v)) \in F$ , arc  $(u, v)$  is contained in *all* shortest  $(u, t)$ -paths with respect to  $\lambda$ , and
- (ii) a forwarding  $F \subseteq V \times A$  is said to be a *unique shortest path forwarding (USPF)* if there exists a compatible metric for  $F$ .

By Observation 4.17, a metric is compatible with a path set  $S \subseteq \mathcal{P}$  if and only if it is compatible with its associated forwarding  $\mathcal{F}(S)$ . Hence, a path set  $S$  is a USPR for the commodity set  $K$  if and only if it contains exactly one  $(s, t)$ -path for each commodity  $(s, t) \in K$  and its associated forwarding  $\mathcal{F}(S)$  is an USPF.

As observed in Chapter 5, both the family of all unique shortest path system as well as the family of all unique shortest path forwarding in a digraph are independence systems. Using a standard integer linear programming formulation for these independence systems, we derive in the following integer linear programming formulations for unsplittable shortest path routing problems without extra variables for the routing metric.

**Remark 7.1** Note that the approach of modeling only the valid end-to-end routing paths and computing a compatible metric afterwards is only applicable if the range of admissible values for the routing metric is sufficiently large. In particular, we assume that we can easily compute a compatible metric with *admissible* length values for any given USPR. For realistic network sizes and modern routing protocols such as OSPF or IS-IS this is normally the case. Otherwise, if the fact that the range of admissible length values is restricted cannot be ignored, further constraints must be added

to the formulations presented in this chapter to ensure the existence of a compatible metric whose arc lengths are admissible for the given routing protocol.

### 7.3 Path Routing Formulations

A natural way to model the unsplittable shortest path routings for a given commodity set in an integer linear programming manner is to use binary variables to indicate which end-to-end paths are contained in the routing and which are not. For this purpose, we introduce the path variables  $x_P \in \{0, 1\}$  for all paths  $P \in \mathcal{P}(K)$ , which are interpreted as

$$x_P = \begin{cases} 1, & \text{if commodity } (s_P, t_P) \text{ is routed via path } P, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

With these variables, we now can define the polytope of all unsplittable shortest path routings.

**Definition 7.2** *The **unsplittable shortest path routing polytope**  $\text{USPR}(K) \subseteq \mathbb{R}^{\mathcal{P}(K)}$  is the convex hull of the characteristic vectors of all unsplittable shortest path routings for the commodity set  $K$ , i.e.,*

$$\text{USPR}(K) := \text{conv}\{ \chi(S) \in \{0, 1\}^{\mathcal{P}(K)} : S \text{ is a USPR for } K \}.$$

Each vertex of  $\text{USPR}(K)$  corresponds to an unsplittable shortest path routing for the commodity set  $K$ , and vice versa. For notational convenience, we denote this polytope simply by  $\text{USPR}$  in the following.

As we have seen in Chapter 5, the family of all unsplittable shortest path systems in  $D$  forms an independence system. Let  $\mathcal{C}_{USPS} \subset 2^{\mathcal{P}}$  be the family of circuits of this independence system. That is,  $\mathcal{C}_{USPS}$  consists of all irreducible non-unsplittable shortest path systems (non-USPS) in  $D$ , see Definition 5.1 on page 72. Clearly, no irreducible non-USPS is fully contained in the path set of any unsplittable shortest path routing. Hence, any vertex  $x$  of  $\text{USPR}$  satisfies the **(USPS) circuit inequalities**

$$\sum_{P \in S} x_P \leq |S| - 1 \quad \forall S \in \mathcal{C}_{USPS} \cap \mathcal{P}(K). \quad (7.2)$$

If it is clear from the context that we are talking about a path routing formulation, we will call these inequalities simply circuit inequalities.

Furthermore, any unsplittable shortest path routing for the commodity set  $K$  contains exactly one  $(s, t)$ -path for each  $(s, t) \in K$ . Consequently, also the **demand constraints**

$$\sum_{P \in \mathcal{P}(s, t)} x_P = 1 \quad \forall (s, t) \in K \quad (7.3)$$

hold for each vertex  $x$  of USPR.

Together with the integrality of the path variables, these constraints are sufficient to describe all unsplittable shortest path routings.

**Theorem 7.3**

$$\text{USPR} = \text{conv}\{x \in \{0, 1\}^{\mathcal{P}(K)} : x \text{ satisfies (7.2) and (7.3)}\}.$$

**Proof.** Clearly, any vertex  $x$  of USPR satisfies  $x \in \{0, 1\}^{\mathcal{P}(K)}$  and all (in)equalities (7.2) and (7.3).

For the reverse direction, let  $x \in \{0, 1\}^{\mathcal{P}(K)}$  be a solution of (7.2)–(7.3), and let  $S(x) := \{P \in \mathcal{P}(K) : x_P = 1\}$  be the set of routing paths defined by this solution.

The circuit inequalities (7.2) ensure that no irreducible non-USPS is fully contained in  $S(x)$ . Thus,  $S(x)$  does not contain any non-USPS, which implies that  $S(x)$  is a USPS.

The demand constraints (7.3) imply that, in addition,  $|S(x) \cap \mathcal{P}(s, t)| = 1$  for all  $(s, t) \in K$ . Consequently,  $S(x)$  is a USPS that contains exactly one comm-path for each commodity  $(s, t) \in K$ . As this means that  $S(x)$  is an unsplittable shortest path routing for the commodity set  $K$ , we have  $x = \chi(S(x)) \in \text{USPR}$ .  $\square$

Note that the set of all circuit inequalities (7.2) contains the inequalities

$$x_P + x_{P'} \leq 1 \quad \forall P, P' \in \mathcal{P}(K) : P \text{ and } P' \text{ are conflicting}$$

as a subset. Taken alone, these inequalities ensures that the path set  $S(x)$  defined by an integer vector  $x \in \{0, 1\}^{\mathcal{P}(K)}$  satisfies the Bellman property.

Theorem 7.3 leads straightforward to integer linear programming formulations for FC-USPR, CAP-USPR, MIN-CON-USPR, or other unsplittable shortest path routing problems.

In order to formulate the fixed charge network design problem FC-USPR, we introduce a binary capacity variable  $z_a \in \{0, 1\}$  for each  $a \in A$ , which is interpreted as

$$z_a = \begin{cases} 1, & \text{if arc } a \text{ is installed, and} \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, an arc set  $B \subseteq A$  provides sufficient capacities for an unsplittable shortest path routing  $S \subseteq \mathcal{P}(K)$  of the commodities  $K$ , if and only if the **capacity constraints**

$$\sum_{(s,t) \in K} \sum_{P \in \mathcal{P}(s,t): a \in P} d_{s,t} x_P \leq u_a z_a \quad \forall a \in A \quad (7.4)$$

hold for the corresponding characteristic vectors  $x = \chi(S)$  and  $z = \chi(B)$ . Hence,

$$\min \{ w^T z : (x, z) \in \{0, 1\}^{\mathcal{P}(K) \cup A}, \quad (FC\text{-}PATHIP) \\ (x, z) \text{ satisfies (7.2)–(7.4)} \}$$

is a correct integer linear programming formulation of the fixed charge network design problem FC-USPR.

Analogously, the capacitated network design problem CAP-USPR can be formulated with integer capacity variables  $z_a \in \mathbb{Z}_+$ ,  $a \in A$ . Each variable  $z_a$  now expresses the number of installed base capacity units on arc  $a \in A$ . With the same capacity constraints (7.4) as for the fixed charge problem (but now with integer capacity variables), the capacitated network design problem CAP-USPR can be formulated as

$$\min \{ w^T z : (x, z) \in \{0, 1\}^{\mathcal{P}(K)} \times \mathbb{Z}_+^A, \quad (CAP\text{-}PATHIP) \\ (x, z) \text{ satisfies (7.2)–(7.4)} \}$$

In order to formulate the congestion minimization problem MIN-CON-USPR, we introduce a single non-negative congestion variable  $L \in \mathbb{R}_+$ . Obviously, the maximum congestion induced by an routing  $S \subseteq \mathcal{P}(K)$  is no more than  $L$  if and only if the **capacity constraints**

$$\sum_{(s,t) \in K} \sum_{P \in \mathcal{P}(s,t): a \in P} d_{s,t} x_P \leq u_a L \quad \forall a \in A \quad (7.5)$$

are satisfied for the corresponding characteristic vectors  $x = \chi(S)$ . Solving the congestion minimization problem MIN-CON-USPR thus is equivalent to solving the mixed integer linear program

$$\min \{ L : (x, L) \in \{0, 1\}^{\mathcal{P}(K)} \times \mathbb{R}_+, \quad (CON\text{-}PATHIP) \\ (x, L) \text{ satisfies (7.2), (7.3), and (7.5)} \}.$$

The correctness of these three (mixed-) integer linear programming models follows immediately from Theorem 7.3. Other unsplittable shortest path routing problems can be formulated analogously.

Note that the circuit inequalities (7.2) are sufficient to ensure that each integer solution of the above formulations indeed defines a valid unsplittable shortest path routing. However, the LPs obtained by relaxing the integrality constraints in these formulations are not very strong. In Chapter 8 we show how additional inequalities to strengthen the formulations can be derived.

Several additional requirements that arise in real-world planning problems can be easily incorporated into these models.

- Path length restrictions, such as upper bounds on the physical length or on the delay of the routing paths, can be encoded by restricting the set of admissible paths  $\mathcal{P}(s, t)$  for each commodity.



- Node or arc taboos, which forbid the use of certain nodes or arcs in the routing paths of some commodities, can be enforced similarly by restricting the admissible path sets.
- Routing symmetry, i.e., the requirement to use of anti-parallel routing paths  $(v_1, v_2, \dots, v_n)$  and  $(v_n, v_{n-1}, \dots, v_1)$  for anti-parallel commodities  $(v_1, v_n)$  and  $(v_n, v_1)$ , can be incorporated into the models either by adding the equalities

$$x_P = x_{P'} \quad \text{for all anti-parallel paths } P, P' \in \mathcal{P}(K),$$

or by performing the corresponding variable substitutions. In fact, the symmetric unsplittable shortest path routings of a given commodity set can be described equivalently by undirected path sets in the corresponding undirected supply graph. Only in the capacity constraints (7.4) or (7.5) the paths need to be considered in a directed context to correctly express the capacity consumption of the directed traffic flows.

In Section 9.1 we present models that include these extensions.

## 7.4 Arc Routing Formulations

Another type of integer linear programming formulations for unsplittable shortest path routings is obtained by describing the end-to-end routing paths in terms of arc routing variables instead of path routing variables.

**Definition 7.4** *For each path set  $S \subseteq \mathcal{P}(K)$ , we denote its corresponding arc routing by*

$$\mathcal{R}(S) := \bigcup_{P \in S} \{ ((s_P, t_P), a) : a \in P \} \subseteq K \times A.$$

In order to indicate which arcs are contained in which routing paths, we introduce a binary arc routing variable  $y_a^{s,t} \in \{0, 1\}$  for each commodity  $(s, t) \in K$  and for each arc  $a \in A$ . These variables are interpreted as

$$y_a^{s,t} = \begin{cases} 1, & \text{if commodity } (s, t) \text{ is routed via arc } a, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

In terms of these variables, the polytope of all valid unsplittable shortest path routings for the commodity set  $K$  is given as follows.

**Definition 7.5** *The **unsplittable shortest path arc routing polytope**  $\text{USPA}(K) \subseteq \mathbb{R}^{K \times A}$  is the convex hull of the incidence vectors of all those arc routings that correspond to an USPR for the commodity set  $K$ , i.e.,*

$$\text{USPA}(K) := \text{conv} \{ \chi(\mathcal{R}(S)) \in \{0, 1\}^{K \times A} : S \text{ is an USPR for } K \}.$$

If it is clear from the context which commodity set  $K$  we refer to, we denote this polytope simply by **USPA** in the following.

Any USPR  $S$  for the commodity set  $K$  contains exactly one  $(s, t)$ -path for each  $(s, t) \in K$ . Hence, its corresponding arc routing variables  $y = \chi(\mathcal{R}(S))$  satisfy the **flow conservation constraints**

$$\sum_{a \in \delta^-(v)} y_a^{s,t} - \sum_{a \in \delta^+(v)} y_a^{s,t} = \begin{cases} -1, & \forall (s, t) \in K, v = s, \\ 1, & \forall (s, t) \in K, v = t, \text{ and} \\ 0, & \forall (s, t) \in K, v \in V \setminus \{s, t\}. \end{cases} \quad (7.6)$$

In order to ensure that the values of the routing variables correspond to a unique shortest path routing, we consider the forwarding defined by their values. Recall that each path set  $S \subseteq \mathcal{P}$  defines its associated forwarding

$$\mathcal{F}(S) = \{(t, a) : a \in P \text{ for some } (s, t)\text{-path } P \in S\} \subseteq V \times A,$$

see Definition 4.16. Similarly, every arc routing defines a forwarding.

**Definition 7.6** *For each arc routing  $R \subseteq K \times A$ , we denote its **associated forwarding** by*

$$\mathcal{F}(R) := \{(t, a) : ((s, t), a) \in R\} \subseteq V \times A.$$

Clearly,  $\mathcal{F}(\mathcal{R}(S)) = \mathcal{F}(S)$  for any path set  $S \subseteq \mathcal{P}$ . Furthermore, a path set  $S$  is a unique shortest path system if and only if its associated forwarding  $\mathcal{F}(S)$  is a unique shortest path forwarding (USPF), see Definition 4.15 and Observation 4.17 on page 61. Hence, a path set  $S$  is a USPS if and only if  $\mathcal{F}(\mathcal{R}(S))$  is a USPF. This condition now can be formulated in terms of the arc routing variables.

As we have seen in Chapter 5, the family of all unique shortest path forwardings (USPF) in  $D$  forms an independence system. Let  $\mathcal{C}_{USPF} \subseteq V \times A$  be the family of circuits of this independence system. By definition,  $\mathcal{C}_{USPF}$  consists of all irreducible non-USPFs in the underlying digraph  $D$ .

The forwarding associated with any unique shortest path system cannot fully contain any non-USPF  $F \in \mathcal{C}_{USPF}$ . Hence, the arc routing variables  $y = \chi(\mathcal{R}(S))$  of an unsplittable shortest path routing  $S$  for the commodity set  $K$  therefore must satisfy the **(USPF) circuit inequalities**

$$\begin{aligned} \sum_{((s,t),a) \in R} y_a^{s,t} &\leq |R| - 1 & \forall R \subseteq K \times A \text{ with} & \quad (7.7) \\ & & (i) \quad \mathcal{F}(R) \in \mathcal{C}_{USPF} & \\ & & (ii) \quad |\mathcal{F}(R)| = |R|. & \end{aligned}$$

Condition (i) in (7.7) ensures that the forwarding associated with each set  $R$  is a non-USPF, which is necessary for (7.7) to be valid for **USPA**. Condition (ii) and the fact that the associated non-USPF  $\mathcal{F}(R)$  is irreducible, on the

other hand, are not necessary for (7.7) to be valid; these two conditions only filter out those sets  $R$  for which inequality (7.7) would be trivially dominated by the variable bounds and other USPF circuit inequalities.

If it is clear from the context that we are talking about an arc routing formulation, we will call these inequalities simply circuit inequalities.

Note that the circuit inequalities (7.7) are just the disaggregated version of the inequalities

$$\sum_{(t,a) \in F} \max\{y_a^{s,t} : s \text{ with } (s,t) \in K\} \leq |F| - 1 \quad \forall F \in \mathcal{C}_{USPF}, \quad (7.7')$$

which involve non-linear max-terms. In fact, the set of all circuit inequalities (7.7) is equivalent to the set of all inequalities (7.7'), see also Section 7.4.1 below.

The following theorem shows that, together with the integrality constraints for the arc routing variables, the linear constraints (7.6) and (7.7) are sufficient to characterize all unsplittable shortest path routings for the commodity set  $K$ .

### Theorem 7.7

$$\text{USPA} = \text{conv}\{y \in \{0,1\}^{K \times A} : y \text{ satisfies (7.6) and (7.7)}\}.$$

**Proof.** First, we show that any vertex of USPA is contained in the polytope defined by the binary vectors  $y \in \{0,1\}^{K \times A}$  that satisfy (7.6) and (7.7). So, let  $y$  be a vertex of USPA. By definition of USPA, there exists an USPR  $S \subset \mathcal{P}(K)$  for the commodity set  $K$ , such that  $y = \chi(\mathcal{R}(S))$ . Obviously,  $y$  is 0/1-valued. It is also easy to see that  $y$  satisfies all equalities (7.6). It remains to show that  $y$  also satisfies all inequalities (7.7).

So, suppose  $y$  violates inequality (7.7) for some set  $R \subseteq K \times A$ . As  $y$  is a 0/1-vector, this can only be the case if  $y_a^{s,t} = 1$  for all  $((s,t),a) \in R$ . For each  $((s,t),a) \in R$ , the arc  $a$  then must be contained in the  $(s,t)$ -path that belongs to the given path set  $S$ . This, however, implies that  $\mathcal{F}(R)$  forms an USPF, which contradicts with condition (i) of inequality (7.7). Consequently,  $y$  satisfies all circuit inequalities (7.7).

For the reverse direction of the proof, consider a vector  $y \in \{0,1\}^{K \times A}$  that satisfies (7.6) and (7.7), and let

$$P_{s,t} := \{a : y_a^{s,t} = 1\} \quad \text{for each } (s,t) \in K.$$

Since  $y$  is binary and fulfills the flow conservation constraints (7.6), each arc set  $P_{s,t}$  contains an  $(s,t)$ -path and maybe some additional circuits. For any directed cycle  $C \subseteq A$  and any  $t \in V$ , however, the set  $\{(t,a) : a \in C\}$  is a non-USPF. The circuit inequalities (7.7) imply that  $y_a^{s,t} = 0$  for at least one arc  $a \in C$ . Consequently, each arc set  $P_{s,t}$  is indeed a simple  $(s,t)$ -path.

Now, suppose that the set  $S := \{P_{s,t} : (s,t) \in K\}$  of these paths does not form an USPR for the commodities  $K$ . According to Observation 4.17 on page 61, its associated forwarding  $\mathcal{F}(S)$  then must be a non-USPF.

Clearly, there exist some irreducible non-USPF  $F \subseteq \mathcal{F}(S)$ . Furthermore, for each  $(t,a) \in F$ , there exists at least one node  $s_{t,a} \in V$  such that  $(s_{t,a}, t) \in K$  and  $y_a^{s_{t,a},t} = 1$ .

One easily verifies that the set  $R := \{((s_{t,a}, t), a) : (t,a) \in F\} \subseteq K \times A$  then satisfies both conditions (i) and (ii) of inequality (7.7). Hence, inequality (7.7) for  $R$  would be violated by the given vector  $y$ . As this is a contradiction to our assumption that  $y$  satisfies all inequalities (7.7), the routing paths  $P_{s,t}$  form an unsplittable shortest path routing for the commodity set  $K$ .  $\square$

Note that the set of all circuit inequalities (7.7) contains the inequalities

$$y_{(u,v)}^{s_1,t} + y_{(u,w)}^{s_2,t} \leq 1 \quad \forall (s_1, t), (s_2, t) \in K, (u, v), (u, w) \in A, v \neq w$$

as a subset. Taken alone, these inequalities ensures that the routing paths of all commodities with a the same destination  $t$  form an anti-arborescence with root  $t$  for any integer vector  $y \in \{0, 1\}^{V \times A}$ .

Theorem 7.7 leads straightforward to another group of (mixed-) integer linear programming formulations for unsplittable shortest path routing problem.

Analogously to the path variable formulations discussed in the previous section, the fixed charge network design problem FC-USPR can be formulated by introducing additional binary capacity variables  $z_a \in \{0, 1\}$ ,  $a \in A$ , and the necessary **capacity constraints**

$$\sum_{(s,t) \in K} d_{s,t} y_a^{s,t} \leq u_a z_a \quad \forall a \in A. \quad (7.8)$$

The complete integer linear programming formulation obtained this way for the fixed charge network design problem FC-USPR is

$$\min \{ w^T z : (y, z) \in \{0, 1\}^{(V \times A) \cup A}, \quad (y, z) \text{ satisfies (7.6)–(7.8)} \}. \quad (\text{FC-ARCIP})$$

Analogously, the capacitated network design problem CAP-USPR can be formulated with general non-negative integer capacity variables  $z_a \in \mathbb{Z}_+$ ,  $a \in A$ . Its corresponding integer linear programming formulation then reads

$$\min \{ w^T z : (y, z) \in \{0, 1\}^{V \times A} \times \mathbb{Z}_+^A, \quad (y, z) \text{ satisfies (7.6)–(7.8)} \}. \quad (\text{CAP-ARCIP})$$

To formulate the congestion minimization problem MIN-CON-USPR, we again introduce a single non-negative congestion variable  $L \in \mathbb{R}_+$  and impose the **capacity constraints**

$$\sum_{(s,t) \in K} d_{s,t} y_a^{s,t} \leq u_a L \quad \forall a \in A \quad (7.9)$$

This yields the following mixed integer linear programming formulation for MIN-CON-USPR:

$$\min \{ L : (y, L) \in \{0, 1\}^{V \times A} \times \mathbb{R}_+, \quad (y, L) \text{ satisfies (7.6), (7.7), and (7.9) } \} . \quad (\text{CON-ARCIP})$$

Theorem 7.7 immediately implies the correctness of these three formulations. Other unsplittable shortest path routing problems can be formulated analogously.

Again, the circuit inequalities (7.7) are sufficient to ensure that each integer solution of the above formulations defines a valid unsplittable shortest path routing, but the LPs obtained by relaxing the integrality constraints are not very strong. Additional inequalities to strengthen the formulations are derived in Chapter 8.

Again, several additional real-world routing restrictions can be easily incorporated into these models.

- Path length restrictions, such as upper bounds on the physical length, the number of arcs, or the delay of the routing paths, can be taken into account by adding inequalities of the type

$$\sum_{a \in A} \ell_a y_a^{s,t} \leq \ell_{s,t}^{\max} \quad \forall (s, t) \in K ,$$

where the coefficients  $\ell_a$  denote the considered length values of the arcs and the numbers  $\ell_{s,t}^{\max}$  denote the imposed length bounds for each commodity  $(s, t) \in K$ .

- Node or arc taboos, which forbid the use of certain nodes or arcs in the routing paths of some commodities, can be enforced by removing the corresponding arc routing variables from the model or, equivalently, setting their upper bounds to 0.
- Routing symmetry, i.e., the use of anti-parallel routing paths for anti-parallel commodities, is easily obtained by either adding the equalities

$$y_{(u,v)}^{s,t} = y_{(v,u)}^{t,s} \quad \forall (s, t), (t, s) \in K, (u, v), (v, u) \in A$$

or by performing the corresponding variable substitutions.

The models presented in Section 9.1 include all these extensions and modifications.

### 7.4.1 Variants with Additional Forwarding Variables

As already mentioned above, the circuit inequalities (7.7) are just the disaggregated form of the simpler but in the strict sense non-linear inequalities (7.7'). By introducing extra variables for the max-expressions involved in inequalities (7.7'), the arc routing formulation can be 'simplified' as follows.

For each node  $t \in V$  and each arc  $a \in A$ , we introduce an additional forwarding variable  $x_a^t \in \{0, 1\}$  to indicate whether the pair  $(t, a)$  is contained in the associated forwarding of the routing. The natural interpretation of these variables is

$$x_a^t = \begin{cases} 1, & \text{if arc } a = (u, v) \text{ is contained in the unique shortest} \\ & (u, t)\text{-path defined by the routing, and} \\ 0, & \text{otherwise.} \end{cases}$$

With these additional variables, the set of circuit inequalities (7.7) can be equivalently formulated as

$$\sum_{(t,a) \in F} x_a^t \leq |F| - 1 \quad \forall F \in \mathcal{C}_{USPF}, \quad (7.7a)$$

$$y_a^{s,t} \leq x_a^t \quad \forall (s, t) \in K, a \in A \quad (7.7b)$$

Inequalities (7.7a) ensure that the forwarding variables describe a unique shortest path forwarding. Together with inequalities (7.6) and (7.7b), this implies that the arc routing variables  $y_a^{s,t}$  describe a valid unsplittable shortest path routing for the commodity set  $K$ .

Consider the polytope associated with this extended integer linear programming formulation.

**Definition 7.8** *The **unique shortest path forwarding polytope**  $USPF(K)$  is the convex hull of all those vectors  $(y, x) \in \{0, 1\}^{(K \times A) \cup (V \times A)}$  that satisfy (in)equalities (7.6), (7.7a), and (7.7b), i.e.,*

$$USPF(K) := \text{conv} \left\{ (y, x) \in \{0, 1\}^{(K \times A) \cup (V \times A)} : \right. \\ \left. (y, x) \text{ satisfies (7.6), (7.7a), (7.7b) } \right\}.$$

If it is clear from the context which commodity set  $K$  we refer to, we again omit the parameter  $K$ .

It is easy to verify the following equivalence.

### Proposition 7.9

$$\begin{aligned} & \left\{ y \in \mathbb{R}_+^{K \times A} : y \text{ satisfies (7.6) and (7.7)} \right\} = \\ & \left\{ y \in \mathbb{R}_+^{K \times A} : \exists x \in \mathbb{R}_+^{V \times A} \text{ s.t. } (y, x) \text{ satisfies (7.6), (7.7a), (7.7b)} \right\} \end{aligned}$$

**Proof.** Follows immediately with  $x_a^t := \max\{y_a^{s,t} : (s,t) \in K\}$  for all  $t \in V$  and  $a \in A$ .  $\square$

Proposition 7.9 implies that the unsplittable shortest path arc routing polytope USPA is equal to the projection of USPF onto the space of arc routing variables  $y$ , i.e.,

$$\text{USPA} = \{y \in \mathbb{R}^{K \times A} : \exists x \in \mathbb{R}^{V \times A} \text{ with } (y, x) \in \text{USPF}\}.$$

Together with Theorem 7.7, this leads to the following result.

**Theorem 7.10** *The polytope  $\text{USPF}(K)$  is the convex hull of all those vectors  $(y, x) \in \{0, 1\}^{(K \times A) \cup (V \times A)}$ , for which  $y$  is the incidence vector of the arc routing of an USPR for the commodity set  $K$  and  $x$  is the incidence vector of an USPF with  $x_a^t \geq y_a^{s,t}$  for all  $(s, t) \in K$  and all  $a \in A$*

Less formally speaking, Theorem 7.10 says that also the extended formulation with additional forwarding variables ‘correctly’ models unsplittable shortest path routings.

Each of the three integer linear programming formulations (FC-ARCIP), (CAP-ARCIP), and (CON-ARCIP) can be transformed into another (mixed-) integer linear programming formulation by adding the  $|V| \cdot |A|$  additional forwarding variables  $x_a^t$  and replacing set of circuit inequalities (7.7) with the set of inequalities (7.7a) and (7.7b). By Theorem 7.10, both alternatives are correct and equivalent with respect to the optimal integer solution value. Even more, Proposition 7.9 implies that also the linear relaxation of the original formulation with inequalities (7.7) and that of the extended formulation with forwarding variables and inequalities (7.7a) and (7.7b) are equivalent, i.e., yield the same optimal solution value.

Also, it is not necessary to require integrality for both the arc routing variables  $y_a^{s,t}$  and for the forwarding variables  $x_a^t$ . One easily verifies that the integrality of all forwarding variables implies the integrality of all arc routing variables. The integrality of all arc routing variables, on the other hand, implies integrality only for those forwarding variables  $x_{(u,v)}^t$  where  $u$  is contained in some chosen  $(s, t)$ -path, but not necessarily for *all* forwarding variables. However, the remaining forwarding variables can be set to 0 without violating any of the inequalities (7.7a) or (7.7b). So, any solution with integer arc routing variables can be transformed into an equivalent solution with both integer arc routing variables and integer forwarding variables.

The fixed charge network design problem FC-USPR thus can be formulated as a mixed integer linear program also in one of the following two ways:

$$\begin{aligned} \min \{ & w^T z : (x, y, z) \in \mathbb{R}_+^{V \times A} \times \{0, 1\}^{(K \times A) \cup A}, \\ & (x, y, z) \text{ satisfies (7.6), (7.7a), (7.7b), and (7.8)} \} \end{aligned} \quad (\text{FC-FWDIP})$$

$$\min \{ w^T z : (x, y, z) \in \{0, 1\}^{V \times A} \times \mathbb{R}_+^{K \times A} \times \{0, 1\}^A, \quad (\text{FC-FWDIP}') \\ (x, y, z) \text{ satisfies (7.6), (7.7a), (7.7b), and (7.8) } \}$$

Both formulations (FC-FWDIP) and (FC-FWDIP') are equivalent to the formulation (FC-ARCIP) with respect to both their optimal integer solution value and their optimal linear relaxation value. Analogously, mixed integer linear programming formulations with forwarding variables and constraints (7.7a) and (7.7b) instead of constraints (7.7) can be derived for FC-USPR, MIN-CON-USPR, or other unsplittable shortest path routing problems.

Similar to the USPF circuit inequalities (7.7), many other valid inequalities for USPA can be formulated easier in terms of the forwarding variables  $x_a^t$  than in terms of the arc routing variables  $y_a^{s,t}$ . The following theorem describes the basic relation between the valid inequalities in both variants.

**Theorem 7.11** *An inequality*

$$\sum_{(t,a) \in V \times A} \alpha_a^t x_a^t \leq \alpha_0$$

with  $\alpha_a^t \geq 0$  for all  $(t, a) \in V \times A$  is valid for USPF if and only if, for all  $R \subseteq K \times A$  with  $\mathcal{F}(R) = V \times A$  and  $|R| = |V| \cdot |A|$ , the inequality

$$\sum_{((s,t),a) \in R} \alpha_a^t y_a^{s,t} \leq \alpha_0$$

is valid for USPA.

**Proof.** Inequalities (7.7b) imply that  $x_a^t \geq \max\{y_a^{s,t} : (s, t) \in K\}$  for all  $t \in V$  and  $a \in A$ . From this, the claim follows straightforward.  $\square$

The remaining two sections of this chapter are dedicated to the questions that naturally arise if one wishes to solve the formulations presented above. In a standard branch-and-bound algorithm, we would iteratively optimize over the respective LP relaxations and branch on one (or a set) of the integer variables whose value is fractional in the current LP solution. From an algorithmic point of view, it thus is natural to ask whether (i) the LP relaxation can be solved efficiently and (ii) which of the alternative formulations has the stronger LP relaxation.

In the following section, we discuss the computational complexity of solving the LP relaxations. Afterwards, in Section 7.6, we address the strength of the relaxations.

## 7.5 Solving the LP Relaxations

In this section, we study the separation and pricing problems associated with the LP relaxations of the integer linear programming formulations introduced above.



In general, all of these formulations involve exponentially many inequalities or variables: The path routing formulations contain both exponentially many path variables  $x_P$ ,  $P \in \mathcal{P}(K)$ , and exponentially many USPS circuit inequalities (7.2), whereas the arc routing formulations contain ‘only’ exponentially many USPF circuit inequalities (7.7) or (7.7a). From the computational perspective, it is an interesting question whether the pricing and separation problems associated with these variables and inequalities can be solved in polynomial time or not.

Note that we actually want to solve the integer linear programs introduced above, and that we intend to do this with the standard LP-based branch-and-bound approach. In the LP relaxations that we then would solve at the nodes of the branch-and-bound tree, some of the commodities routing variables may be fixed already. When solving an LP in the branch-and-bound tree, these fixings need to be considered. Since we consider unsplittable shortest path routings, there are interdependencies among all paths of all commodities. Even the routing paths that are entirely fixed by the previous branching decisions must be explicitly considered in the LP relaxation of a subproblem; they still may imply constraints on the other paths and thus lead to the separation of new inequalities.

In order to keep notation and argumentation as simple as possible, we focus on the integer programming formulations for the fixed charge network design problem FC-USPR in this and in the following section. The obtained results carry over straightforward to the formulations of CAP-USPR and MIN-CON-USPR.

### 7.5.1 Path Routing Formulations

The formulation (FC-PATHIP) contains exponentially many path routing variables  $x_P$ ,  $P \in \mathcal{P}(K)$ , as well as exponentially many USPS circuit inequalities (7.2) in general. It therefore is natural to solve its LP relaxation using a price-and-cut algorithm, which iteratively solves a series of so-called *restricted LP relaxations*.

In principle, a price-and-cut algorithm works as follows: It starts with a restricted LP (RLP) that consists of only a small initial subset of the variables and constraints of the entire LP relaxation. In each iteration, it first solves the current (RLP) and then determines some path variables with negative reduced cost and/or some violated inequalities. These are then added to (RLP) for the next iteration. The algorithm stops if neither variables with negative reduced costs nor violated inequalities are found. The optimal solution of the last iteration’s (RLP) is optimal and feasible for the complete LP relaxation. Barnhart et al. [16, 18, 17] and Desrosiers and Lübbecke [70, 71] discuss theoretical aspects and successful implementations of such algorithms.

The efficiency of the overall price-and-cut algorithm depends substantially

on how efficient variables with negative reduced costs and violated inequalities can be found. In our case, the separation problem for USPS circuit inequalities is  $\mathcal{NP}$ -hard, unfortunately. Even more, there exists a class of problem instances for which the LP relaxation of (FC-PATHIP) has a unique optimal basis with exponentially many active path variables and exponentially many binding circuit inequalities. We will present such an example at the end of this section. Hence, the LP relaxation of (FC-PATHIP) cannot be solved in polynomial time in general.

On the positive side, we will show that the separation problem for the circuit inequalities (7.2) can be solved in polynomial time if the given vector  $x$  is integer, and we will show that, in each iteration of the price-and-cut algorithm, the pricing problem for the path variables  $x_P$ ,  $P \in \mathcal{P}$ , can be solved in polynomial time with respect to size of the current restricted LP.

### Separation of USPS Circuit Inequalities

First, we consider the separation problem for the USPS circuit inequalities (7.2). Formally, this separation problem is given as follows:

<i>Problem:</i>	SEP-USPS-CIRCUIT
<i>Instance:</i>	A vector $x \in \mathbb{Q}^{\mathcal{P}(K)}$ .
<i>Task:</i>	Find an irreducible non-USPS $S \in \mathcal{C}_{USPS}$ for which $x$ violates the circuit inequality (7.2) or prove that no such $S$ exists.

Without loss of generality, we may assume that the fractional solution  $x$  satisfies at least the trivial variable bounds, i.e.,  $x \in [0, 1]^{\mathcal{P}(K)}$ . Furthermore, we may assume that  $x$  is given as a sparse-vector, i.e., only the non-zero entries of  $x$  are actually encoded. In other words, the encoding size of the separation problem SEP-USPS-CIRCUIT depends only on the non-zero values in  $x$ , but not on the size of the entire path set  $S$ .

For any given such vector  $x$  and any irreducible non-USPS  $S$ , one can easily verify in polynomial time whether the corresponding circuit inequality (7.2) is violated or not. Therefore, SEP-USPS-CIRCUIT is an  $\mathcal{NP}$  decision problem. With the results of Chapter 5, it follows that SEP-USPS-CIRCUIT is also  $\mathcal{NP}$ -hard.

**Theorem 7.12** *SEP-USPS-CIRCUIT is  $\mathcal{NP}$ -complete in general.*

**Proof.** Given a vector  $x \in [0, 1]^{\mathcal{P}(K)}$ , SEP-USPS-CIRCUIT is equivalent to the problem of finding an irreducible non-USPS  $S \in \mathcal{C}_{USPS}$  with  $\sum_{P \in S} (1 - x_P) < 1$  or proving that no such  $S$  exists. This, in turn, is equivalent to solving the MIN-WEIGHT-NON-USPS problem for the path set  $S' := \{P : x_P > 0\}$  and the path weights  $w_P := (1 - x_P)$  for all  $P \in S$ . The latter problem was shown to be  $\mathcal{NP}$ -hard to approximate within a factor strictly

less than  $7/6$  in Theorem 5.15 and Corollary 5.16. Hence, SEP-USPS-CIRCUIT is  $\mathcal{NP}$ -hard as well.  $\square$

One can show that SEP-USPS-CIRCUIT remains  $\mathcal{NP}$ -hard even if  $x$  is given by an optimal solution  $(x, z)$  of some restricted LP relaxation of (FC-PATHIP).

Because SEP-USPS-CIRCUIT is  $\mathcal{NP}$ -hard, we cannot hope to solve the LP relaxation of (FC-PATHIP) in polynomial time (unless  $\mathcal{P} = \mathcal{NP}$ ). However, we can efficiently cut-off *integer* vectors  $x$  that do not correspond to unsplittable shortest path routings.

**Theorem 7.13** *SEP-USPS-CIRCUIT can be solved in polynomial time for binary vectors  $x \in \{0, 1\}^{\mathcal{P}(K)}$ .*

**Proof.** Let  $x \in \{0, 1\}^{\mathcal{P}(K)}$  and  $S(x) := \{P \in \mathcal{P}(K) : x_P = 1\}$ . Then SEP-USPS-CIRCUIT is equivalent to finding an irreducible non-USPS  $R \subseteq S(x)$  or proving that  $S(x)$  is a USPS. This problem can be solved in polynomial time with respect to  $|S(x)|$  using algorithm Greedy-Non-USPS presented in Chapter 5, see Theorem 5.13 on page 83. Because  $|S(x)| \leq |K|$  for any  $x \in \{0, 1\}^{\mathcal{P}(K)}$  that satisfies (7.3), SEP-USPS-CIRCUIT is solved in polynomial time with respect to  $|K|$ .  $\square$

Note that Theorem 7.13 only implies that one can find *some* violated circuit inequality (7.2) for any invalid integer vector  $x$ . Finding the most violated such inequality (in the normalized form with right-hand side 1) is equivalent to the problem of finding the minimum cardinality non-USPS  $R \subseteq S(x)$ , which again is  $\mathcal{NP}$ -hard, see Theorem 5.15.

There are several special cases, in which also the general separation problem SEP-USPS-CIRCUIT can be solved in polynomial time. One such case comprises symmetric (or undirected) unsplittable shortest path routing problems where the underlying graph is a cycle or, more general, belongs to the class of graphs described in Theorem 5.26 on page 100. In these graphs, the family of unsplittable shortest path routings is completely characterized by the Bellman property. Therefore, the size of any irreducible non-USPSs is exactly two, so SEP-USPS-CIRCUIT can be solved in polynomial time by enumerating all possible irreducible non-USPSs.

### Pricing of Path Variables

In the sequel, we turn our attention to the pricing problem for the path variables  $x_P$ ,  $P \in \mathcal{P}(K)$ .

Consider the LP relaxation of (FC-PATHIP) and let  $\eta_S$ ,  $S \in \mathcal{C}_{USPS}$ , be the dual variables of the circuit inequalities (7.2), let  $\pi_{s,t}$ ,  $(s, t) \in K$ , be the dual variables of the demand constraints (7.3), and let  $\mu_a$ ,  $a \in A$ , be the dual variables of the capacity constraints (7.4). The problem of finding a path variable with negative reduced cost then can be formulated as follows:

<i>Problem:</i>	PRICE-USPR-PATH
<i>Instance:</i>	Vectors $\eta \in \mathbb{Q}^{\mathcal{C}_{USPS}}$ , $\pi \in \mathbb{Q}^K$ , and $\mu \in \mathbb{Q}^A$ .
<i>Task:</i>	Find an $(s, t)$ -path $P \in \mathcal{P}(K)$ such that $-\pi_{s,t} - \sum_{a \in P} \mu_a - \sum_{S \in \mathcal{C}_{USPS}: P \in S} \eta_S < 0 \quad (7.10)$ or show that no such path exists.

In view of this problem, we may assume without loss of generality that  $\eta_S \geq 0$  for all  $S \in \mathcal{C}_{USPS}$ , that  $\pi_{s,t} \leq 0$  for all  $(s, t) \in K$ , and that  $\mu_a \geq 0$  for all  $a \in A$ .

It is easy to see the PRICE-USPR-PATH is in  $\mathcal{NP}$ . We also have reason to believe that, for general vectors  $\eta$ ,  $\pi$ , and  $\mu$ , this problem is  $\mathcal{NP}$ -hard.

**Conjecture 7.14** *For arbitrary vectors  $\eta \in \mathbb{Q}_+^{\mathcal{C}_{USPS}}$ ,  $\pi \in \mathbb{Q}_-^K$ , and  $\mu \in \mathbb{Q}_+^A$ , PRICE-USPR-PATH is  $\mathcal{NP}$ -hard.*

However, within a branch-and-price-and-cut algorithm for (FC-PATHIP) the vectors  $\eta$ ,  $\pi$ , and  $\mu$  are not arbitrary. They comprise an optimal dual solution of some restricted LP relaxation of (FC-PATHIP).

In the following we show that, given a restricted LP relaxation (RLP) of (FC-PATHIP), the problem of finding a path variable with negative reduced cost can be solved in polynomial time with respect to the encoding size of (RLP). In this setting, we may assume without loss of generality that also the set  $R \subseteq \mathcal{P}(K)$  of all path variables contained in (RLP) is given as input to the pricing problem PRICE-USPR-PATH.

The following lemma shows that we can restrict our attention to those vectors  $\eta$ ,  $\pi$ , and  $\mu$  that satisfy

$$\eta_S = 0 \quad \text{for all } S \in \mathcal{C}_{USPS} \text{ with } S \not\subseteq R. \quad (7.11)$$

**Lemma 7.15** *Let  $R \subseteq \mathcal{P}(K)$  and consider a restricted LP relaxation (RLP) of (FC-PATHIP). Assume that*

- (i) *(RLP) contains all capacity variables  $z_a$ ,  $a \in A$ ,*
- (ii) *(RLP) contains exactly those path variables  $x_P$  with  $P \in R$ ,*
- (iii) *(RLP) contains the variable bound constraints  $0 \leq x_P \leq 1$  for all  $P \in R$ , and*
- (iv) *(RLP) contains all (in)equalities (7.3) and (7.4).*

*Furthermore, (RLP) may contain an arbitrary number of circuit inequalities (7.2).*

*Then there exists an optimal dual solution  $(\eta^*, \pi^*, \mu^*)$  of (RLP) that satisfies (7.11).*

**Proof.** For all those  $S \in \mathcal{C}_{USPS}$ , for which the corresponding circuit inequality (7.2) is not contained in (RLP), we trivially have  $\eta_S = 0$  for any dual feasible solution  $(\eta, \pi, \mu)$  of (RLP).

So let  $S \in \mathcal{C}_{USPS}$  be an irreducible non-USPS with  $S \not\subseteq R$ , whose corresponding circuit inequality (7.2) is contained in (RLP).

The path variables  $x_P$  with  $P \in S \setminus R$  are not contained in (RLP), so we have  $x_P = 0$  for all  $P \in S \setminus R$  in any solution  $(x, z)$  of (RLP). (Every solution  $(x, z)$  of (RLP) is interpreted as a solution  $(x, z)$  of the entire linear relaxation of (FC-PATHIP) with  $x_P = 0$  for all  $P \notin R$ .) For each other path  $P \in S \cap R$ , (RLP) contains the path routing variable  $x_P$  together with its corresponding variable bound constraint  $x_P \leq 1$ . As  $S \not\subseteq R$ , these variable bound constraints imply

$$\sum_{P \in S} x_P = \sum_{P \in S \cap R} x_P + \sum_{P \in S \setminus R} x_P = \sum_{P \in S \cap R} x_P \leq |S| - 1.$$

Consequently, the circuit inequality (7.2) for  $S$  is redundant in (RLP).

In fact, the above argument shows that (7.2) is redundant for all path sets  $S \in \mathcal{C}_{USPS}$  with  $S \not\subseteq R$ . It then follows immediately from linear programming duality that there exists a dual optimal solution  $(\eta^*, \pi^*, \mu^*)$  with  $\eta_S^* = 0$  for all these path sets  $S$ .  $\square$

Using a simple perturbation technique when solving (RLP), one can easily enforce that the optimal dual solution  $(\eta^*, \pi^*, \mu^*)$  of (RLP), which is considered as input for the pricing problem, satisfies (7.11).

**Remark 7.16** Lemma 7.15 implies that there exist optimal dual multipliers satisfying (7.11) even if the restricted LP (RLP) contains all circuit inequalities (7.2) of (FC-PATHIP). However, if the LP relaxation of (FC-PATHIP) is solved by a price-and-cut algorithm as proposed at the beginning of this section, then the intermediate restricted LPs do not even contain circuit inequalities (7.2) whose underlying non-USPS  $S$  is not fully contained in the set of active path variables. Lemma 7.15 and property (7.11) then hold trivially.

Property (7.11) implies that, for each path  $P \not\subseteq R$ , we have  $\eta_S = 0$  for all  $S \in \mathcal{C}_{USPS}$  with  $P \not\subseteq R$   $P \in S$ . Given the set  $R \subseteq \mathcal{P}(K)$  of path contained in (RLP) and a dual feasible solution  $(\eta, \pi, \mu)$  of (RLP) that satisfies (7.11), PRICE-USPR-PATH thus reduces to the problem of finding a path  $P \subset \mathcal{P}(K) \setminus R$  that satisfies

$$-\pi_{s,t} - \sum_{a \in P} \mu_a < 0, \tag{7.12}$$

or to prove that no such path exists.

This restricted version of PRICE-USPR-PATH can be solved easily by computing the  $(|R \cap \mathcal{P}(s, t)| + 1)$  shortest  $(s, t)$ -paths with respect to the arc

lengths  $\mu_a$ ,  $a \in A$ , for each  $(s, t) \in K$ . If the length of the shortest  $(s, t)$ -path that does not belong to  $R$  is less than  $-\pi_{s,t}$ , then this path satisfies (7.12), and consequently also (7.10). Otherwise no  $(s, t)$ -path with negative reduced cost exists.

For each commodity  $(s, t) \in K$ , the  $(|R \cap \mathcal{P}(s, t)| + 1)$  shortest  $(s, t)$ -paths can be computed in polynomial time with respect to  $|R \cap \mathcal{P}(s, t)|$  and to the encoding size of  $\mu$ , using the  $k$ -shortest paths algorithms of Yen [196] for example.

This immediately yields the following theorem.

**Theorem 7.17** *Let  $(RLP)$  be a restricted LP relaxation of (FC-PATHIP). Furthermore, let  $R \subseteq \mathcal{P}(K)$  be the set of all paths whose corresponding path variable is contained in  $(RLP)$  and let  $(\pi, \eta, \mu)$  be a dual solution of  $(RLP)$  that satisfies (7.11). Then PRICE-USPR-PATH can be solved in polynomial time with respect to  $|R|$ ,  $|A|$ , and the encoding size of  $\eta$ ,  $\pi$ , and  $\mu$ .*

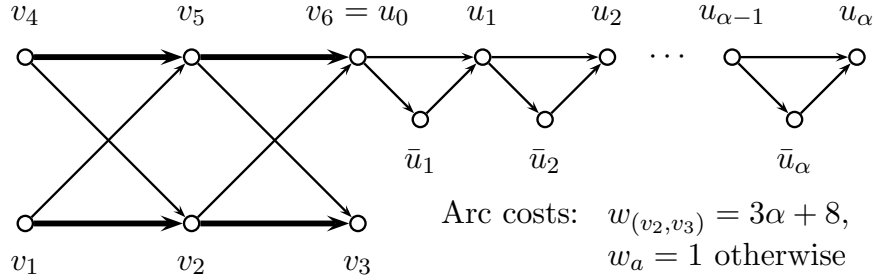
Theorem 7.17 implies that, in each iteration of the price-and-cut algorithm, a single path variable with negative reduced cost can be found in polynomial time with respect to the size of the current restricted LP. Unfortunately, the process of iteratively adding violated inequalities or missing path variables with negative reduced cost does not necessarily terminate after only polynomially many iterations, as we will see in the following section.

**Remark 7.18** Note that the pricing problem remains polynomially solvable if some nodes or arcs are forbidden in the routing paths of some commodities or if path length restrictions based on uniform arc lengths are introduced. To cope with the first kind of restrictions, the  $k$ -shortest paths problem can be solved in the corresponding subgraph of  $D$ . If length bounds are imposed on the admissible routing paths, the pricing problem turns into a  $\ell$ -bounded  $k$ -shortest paths problem, which is still solvable in polynomial time with a variant of Yen's algorithm [196] for uniform arc lengths. If the underlying arc lengths are arbitrary, the  $\ell$ -bounded  $k$ -shortest paths problem and our pricing problem become (weakly)  $\mathcal{NP}$ -hard.

Node and link taboos or limits on the hop-length of the paths, as introduced in our real models, therefore do not affect the solvability of the pricing problem. Also, branching schemes that work by fixing the path variables of all paths that cross a chosen set of arcs to 0, like the set-based branching scheme introduced by Barnhart et al. [18], do not affect the nature of the pricing problem. These fixings can be handled easily by reducing the graph used in the  $k$ -shortest paths computation, again.

### Exponentially Large Optimal Bases

In general, an optimal basis of the full LP relaxation at some node of the branch-and-bound tree may involve an exponential number of variables and



**Figure 7.1** Digraph  $D$  with indicated capacity bounds and costs: Bold arcs have a capacity of 2, thin arcs have a capacity of 1.

constraints (with respect to the encoding size of the underlying FC-USPR problem).

**Proposition 7.19** *There exist classes of FC-USPR instances, for which the optimal basis of the LP relaxation of (FC-PATHIP) is unique and contains both an exponential number of active path variables  $x_P$  and an exponential number of binding circuit inequalities (7.2).*

**Proof.** In order to simplify and shorten the presentation, we provide a class of examples where the routing paths of some commodities are already fixed. If (FC-PATHIP) is solved by an LP-based branch-and-bound algorithm, this is the case after some branches anyway. It is not difficult to extend this class of examples to the case without fixed routing paths.

Let  $\alpha \in \mathbb{Z}_+$  and consider the digraph  $D = (V, A)$  with capacities  $u_a$  and arc costs  $w_a$  illustrated in Figure 7.1. The node set  $V$  consist of the  $6 + 2\alpha$  nodes  $v_1, \dots, v_6, u_1, \dots, u_\alpha$ , and  $\bar{u}_1, \dots, \bar{u}_\alpha$ . The arc set  $A$  consists of the 8 arcs  $(v_1, v_2), (v_2, v_3), (v_4, v_5), (v_5, v_6), (v_1, v_5), (v_5, v_3), (v_4, v_2), (v_2, v_6)$  and the  $3\alpha$  arcs  $(u_{i-1}, u_i), (u_{i-1}, \bar{u}_i)$ , and  $(\bar{u}_i, u_i)$  with  $i = 1, \dots, \alpha$ , where  $u_0 := v_6$ . The capacities  $u_a$  and the arc costs  $w_a$  are set to

$$u_a := \begin{cases} 2, & \text{for } a \in \{(v_1, v_2), (v_2, v_3), (v_4, v_5), (v_5, v_6)\}, \\ 1, & \text{otherwise,} \end{cases}$$

and

$$w_a := \begin{cases} 3\alpha + 8, & \text{for } a = (v_2, v_3), \\ 1, & \text{otherwise.} \end{cases}$$

Note that size of the graph as well as the encoding size of the capacity bounds and arc costs are polynomial in  $\alpha$ .

Among the nodes of  $D$  we consider the four commodities  $(v_1, v_3), (v_4, v_3), (v_4, v_6)$ , and  $(v_1, u_\alpha)$  with demand values

$$d_{s,t} := \begin{cases} 2, & \text{for } (s, t) = (v_4, v_6), \text{ and} \\ 1, & \text{for } (s, t) \in \{(v_1, v_3), (v_4, v_3), (v_1, u_\alpha)\}. \end{cases}$$

We assume that the routing paths of the commodities  $(v_4, v_3)$  and  $(v_4, v_6)$  have already been fixed by the Branch-and-Cut-and-Price algorithm to (the only feasible choice)

$$\begin{aligned} P_{4,3} &:= (v_4, v_2, v_3) \quad \text{for commodity } (v_4, v_3) \text{ and} \\ P_{4,6} &:= (v_4, v_5, v_6) \quad \text{for commodity } (v_4, v_6). \end{aligned}$$

In the following, we show that the LP relaxation of (FC-PATHIP) has a unique optimal solution  $(x^*, z^*)$  satisfying

$$x_{P_{4,3}} = x_{P_{4,6}} = 1, \quad (7.13)$$

and that  $x_P^* > 0$  for exponentially many paths  $P$ .

First, let us see which path variables  $x_P$  may have positive values in such a solution. Commodity  $(v_1, v_3)$  may be routed fractionally over both of the two  $(v_1, v_3)$ -paths  $P_{1,3}^A := (v_1, v_5, v_3)$  and  $P_{1,3}^B := (v_1, v_2, v_3)$ . Note that the capacity of  $(v_5, v_6)$  is already saturated by routing commodity  $(v_4, v_6)$  across this arc. For commodity  $(v_1, u_\alpha)$  therefore only those  $(v_1, u_\alpha)$ -paths  $P$  with  $(v_5, v_6) \notin P$  are eligible. One easily verifies that there are exactly  $2^\alpha$  many such  $(v_1, u_\alpha)$ -paths, and we denote these paths  $P_{1,\alpha}^1$  to  $P_{1,\alpha}^{2^\alpha}$ .

Next, observe that each of these paths  $P_{1,\alpha}^i$  contains the subpath  $(v_1, v_2, v_6)$ . Hence, every path set  $S_i := \{P_{1,3}^A, P_{4,3}, P_{4,6}, P_{1,\alpha}^i\}$ ,  $i = 1, \dots, 2^\alpha$ , is an irreducible non-USPS. Consequently, (FC-PATHIP) contains a circuit inequality (7.2) for each such set  $S_i$ . All other inequalities (7.2) in (FC-PATHIP) are redundant for the subproblem defined by (7.13).

Furthermore, the demand constraints (7.3) imply that any  $x$  that satisfies (7.13) also fulfills

$$\max\{x_{P_{1,\alpha}^i} : i = 1, \dots, 2^\alpha\} \geq 2^{-\alpha}, \quad (7.14)$$

$$x_{P_{1,3}^A} \leq 1 - 2^{-\alpha} \quad \text{and} \quad (7.15)$$

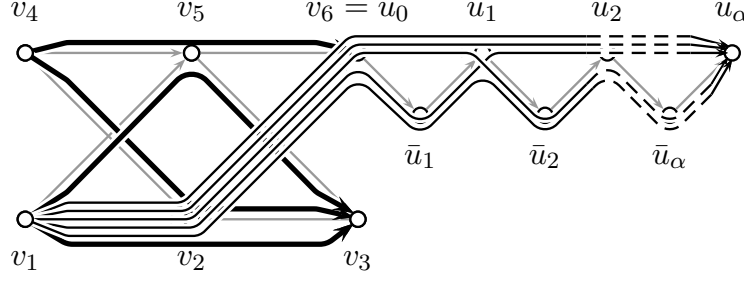
$$x_{P_{1,3}^B} \geq 2^{-\alpha}. \quad (7.16)$$

Clearly, equality holds in (7.14), (7.15), and (7.16) if and only if  $x_{P_{1,\alpha}^i} = 2^{-\alpha}$  for all  $i = 1, \dots, 2^\alpha$ .

Note that we have  $w_{(v_2, v_3)} > \sum_{a \neq (v_2, v_3)} w_a$  for the arc costs defined in our example. Hence, there is a unique optimal fractional solution  $(x^*, z^*)$  for (FC-PATHIP) satisfying (7.13), which is given by

$$x_P^* = \begin{cases} 1, & \text{for } P = P_{4,3} \text{ and } P = P_{4,6}, \\ 1 - 2^{-\alpha}, & \text{for } P = P_{1,3}^A, \\ 2^{-\alpha}, & \text{for } P = P_{1,3}^B \text{ and } P \in \{P_{1,\alpha}^i : i = 1, \dots, 2^\alpha\}, \text{ and} \\ 0, & \text{otherwise,} \end{cases}$$





**Figure 7.2** Optimal fractional solution of (FC-PATHIP) with exponentially many active variables and inequalities.

and

$$z_a^* = \begin{cases} 1, & \text{for } a \in \{(v_4, v_5), (v_5, v_6), (v_4, v_2), (v_2, v_6)\}, \\ (1 + 2^{-\alpha})/2, & \text{for } a = (v_1, v_2) \text{ and } a = (v_2, v_3) \\ 1 - 2^{-\alpha}, & \text{for } a = (v_1, v_5) \text{ and } a = (v_5, v_3), \text{ and} \\ 1/2, & \text{otherwise.} \end{cases}$$

Figure 7.2 illustrates the routing corresponding to this solution.

In the corresponding basis, obviously all  $2^\alpha$  path variables  $x_{P_{1,\alpha}^i}$  are active. It is not difficult to verify that each of the  $2^\alpha$  circuit inequalities (7.2) for the sets  $S_i$  is tight and active in this basis. If one of these inequalities were remove from the LP, then some  $(v_1, v_3)$ -flow could be shifted from path  $P_{1,3}^B$  to path  $P_{1,3}^A$ , leading to less capacity installation on arc  $(v_2, v_3)$  and a better objective value.

With  $\alpha \rightarrow \infty$  Proposition 7.19 follows.  $\square$

Proposition 7.19 implies that we cannot expect to solve the LP relaxation of (FC-PATHIP) in polynomial time.

**Remark 7.20** In the simplified example shown above one could easily lift all path variables  $P_{1,\alpha}^i$ ,  $i = 1, \dots, 2^\alpha$ , into the circuit inequality (7.2) of some set  $S_j$ . This would yield the general rank inequality

$$\sum_{i=1, \dots, 2^\alpha} x_{P_{1,\alpha}^i} + x_{P_{1,3}^A} + x_{P_{4,3}} + x_{P_{4,6}} \leq 3,$$

which dominates all  $2^\alpha$  circuit inequalities (7.2) for the sets  $S_j$ ,  $j = 1, \dots, 2^\alpha$ . The LP including this inequality would have an optimal basis of polynomial size for the case discussed above. However, it is not clear how to find a set of valid inequalities such that the resulting optimal basis is only polynomially large in general.

**Remark 7.21** Proposition 7.19 and the presented class of examples carry over straightforward to the LP relaxation of (CON-PATHIP). It also hold for

the restricted LP relaxations of (CAP-PATHIP) that are solved at the sub-nodes of an LP-based branch-and-bound algorithm, where some of the commodities routing paths or some of the capacity variables are already fixed. However, it does not hold for the plain LP relaxation of (CAP-PATHIP) without all variables free. This LP can be solved easily by routing each commodity unsplit along a shortest path with respect to the metric  $w_a/u_a$  for all  $a \in A$ .

### 7.5.2 Arc Routing Formulations

In contrast to the formulation based on path routing variables, the formulations based on arc routing variables contain ‘only’ a polynomial number of variables. Yet, also these formulations involve exponentially many (non-redundant) non-USPF inequalities of type (7.7) or (7.7a).

In the following, we show that the separation problem for these inequalities is  $\mathcal{NP}$ -hard. With the polynomial time equivalence of separation and optimization [103], this implies that solving the LP relaxation of (FC-ARCIP) is  $\mathcal{NP}$ -hard as well. On the other hand, we again can cut-off infeasible integer vectors in polynomial time.

For both types of USPF circuit inequalities (7.7) and (7.7a), the task of finding a violated inequality leads to the same combinatorial problem, which can be formalized as follows.

<i>Problem:</i>	SEP-USPF-CIRCUIT
<i>Instance:</i>	An arbitrary vector $x \in \mathbb{Q}^{V \times A}$ .
<i>Task:</i>	Find an irreducible non-USPF $F \in \mathcal{C}_{USPF}$ such that $\sum_{(t,a) \in F} x_a^t >  F  - 1$ or prove that no such $F$ exists.

To find a violated circuit inequality of type (7.7a) in the formulation that contains both arc routing and forwarding variables, we need to solve SEP-USPF-CIRCUIT for the vector  $x$  given by the values of the forwarding variables  $x_a^t$  in the current LP solution. Obviously, inequality (7.7a) is violated for an irreducible non-USPF  $F$  if and only if  $\sum_{(t,a) \in F} x_a^t > |F| - 1$ .

In order to find a violated circuit inequality of type (7.7) in the formulation without forwarding variables, we solve SEP-USPF-CIRCUIT for the vector  $x$  given by

$$x_a^t := \max \{ y_a^{s,t} : (s,t) \in K \} \quad \text{for all } t \in V \text{ and } a \in A,$$

where  $y_a^{s,t}$  are the values of the arc routing variables in the current LP solution. It is easy to see that there exist an irreducible non-USPF  $F$  with

$\sum_{(t,a) \in F} x_a^t > |F| - 1$  for this vector  $x$ , if and only if inequality (7.7) is violated for the arc routing

$$R := \{((s_{t,a}, t), a) : (t, a) \in F \text{ and } s_{t,a} = \arg \max\{y_a^{s,t} : (s, t) \in K\}\}.$$

In either case we may assume without loss of generality that the given fractional solution satisfies at least the variables bound constraints and, therefore,  $x \in [0, 1]^{V \times A}$ .

It follows from the results in Chapter 5 that SEP-USPF-CIRCUIT is  $\mathcal{NP}$ -complete.

**Theorem 7.22** SEP-USPF-CIRCUIT is  $\mathcal{NP}$ -complete in general.

**Proof.** Given a vector  $x \in [0, 1]^{V \times A}$  with  $x_a^t \in \mathbb{Q}$  for all  $t \in V$  and all  $a \in A$  and given an irreducible non-USPF  $S$ , we obviously can verify in polynomial time whether  $\sum_{(t,a) \in F} x_a^t > |F| - 1$  or not. So SEP-USPF-CIRCUIT belongs to the class of  $\mathcal{NP}$  decision problems.

On the other hand, SEP-USPF-CIRCUIT is equivalent to the problem of finding an irreducible non-USPF  $F \in V \times A$  with  $\sum_{(t,a) \in F} (1 - x_a^t) < 1$  or proving that no such  $F$  exists. This problem, in turn, is equivalent to solving the MIN-WEIGHT-NON-USPF problem for the weights  $w_{t,a} := (1 - x_a^t)$  for all  $t \in V$  and all  $a \in A$ . By Corollary 5.33 on page 109, the later problem is  $\mathcal{NP}$ -hard to approximate within a factor strictly less than  $7/6$ . Hence, SEP-USPF-CIRCUIT is  $\mathcal{NP}$ -hard in general.  $\square$

SEP-USPF-CIRCUIT remains  $\mathcal{NP}$ -hard even if the given vector  $x$  corresponds an optimal basic solution of a (restricted) LP of the original formulation.

Again, we can solve the separation problem in polynomial time if the given vector  $x$  is integer.

**Theorem 7.23** SEP-USPF-CIRCUIT can be solved in polynomial time for binary vectors  $x \in \{0, 1\}^{V \times A}$ .

**Proof.** Given a binary vector  $x \in \{0, 1\}^{V \times A}$ , let  $F(x) := \{(t, a) : x_a^t = 1\}$ . The separation problem SEP-USPF-CIRCUIT then is equivalent to the problem of finding an irreducible non-USPF  $E \subseteq F(x)$  or proving that no such non-USPF exists. This problem can be solved in polynomial time using the greedy algorithm Greedy-Non-USPF presented in Chapter 5, see Theorem 5.31 on page 106.  $\square$

Theorem 7.23 implies that we can find *some* violated circuit inequality (7.7) or (7.7a) if the given integer vector  $x \in \{0, 1\}^{V \times A}$  does not correspond to an unsplittable shortest path routing. Finding the most violated such inequality (in the normalized form with right-hand side 1) is equivalent to the problem of finding the minimum cardinality non-USPF  $E \subseteq F(x)$ . This problem again is  $\mathcal{NP}$ -hard, see Theorem 5.15 on page 84.

In the special cases where the rank of all irreducible non-USPFs in the underlying graph is bounded by some constant  $k \in \mathbb{Z}_+$  or where the unique shortest path forwardings are completely characterized by the Bellman property, SEP-USPS-CIRCUIT can be solved in polynomial time using enumerative methods. The class of symmetric (or undirected) unsplittable shortest path routing problems where the underlying undirected graph is a cycle or, more general, belongs to the class of graphs described in Theorem 5.26 on page 100 is such as case.

## 7.6 Strength of the LP Relaxations

In the following section, we finally compare the strength of the three alternative (mixed-) integer linear programming formulations (FC-PATHIP), (FC-ARCIP), and (FC-FWDIP).

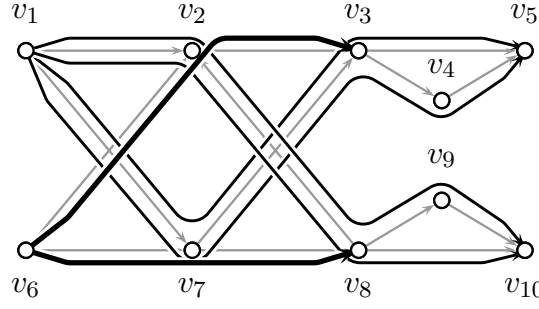
For this, let  $\text{USPR}_z$ ,  $\text{USPA}_z$ , and  $\text{USPF}_z$  be the three polytopes obtained by projecting the solutions of the LP relaxations of (FC-PATHIP), (FC-ARCIP), and (FC-FWDIP) onto the space of capacity variables  $z_a$ ,  $a \in A$ , i.e.,

$$\begin{aligned} \text{USPR}_z &:= \text{conv} \left\{ z \in [0, 1]^A : \exists x \in [0, 1]^{\mathcal{P}(K)} \text{ such that} \right. \\ &\quad \left. (x, z) \text{ satisfies (7.2)–(7.4)} \right\}, \\ \text{USPA}_z &:= \text{conv} \left\{ z \in [0, 1]^A : \exists y \in [0, 1]^{K \times A} \text{ such that} \right. \\ &\quad \left. (y, z) \text{ satisfies (7.6)–(7.8)} \right\}, \text{ and} \\ \text{USPF}_z &:= \text{conv} \left\{ z \in [0, 1]^A : \exists (x, y) \in [0, 1]^{(K \times A) \cup (V \times A)} \text{ such that} \right. \\ &\quad \left. (y, x, z) \text{ satisfies (7.6), (7.7a), (7.7b), (7.8)} \right\}. \end{aligned}$$

In order to see whether the LP relaxation of one formulation is stronger, equivalent, or incomparable to that of another formulation, we need to compare these polytopes. If two of these polytopes coincide for all instances of FC-USPR, then the respective formulations are equally strong. If, for all instances of FC-USPR, one polytope is fully contained in another one, then the formulation corresponding to the smaller polytope is always at least as strong as that corresponding to the larger one. Finally, if there exist a digraph  $D = (V, A)$  with arc capacities  $u_a$ ,  $a \in A$ , and a commodity set  $K \subseteq V \times V$  with demand values  $d_{s,t}$ ,  $(s, t) \in K$ , such that none of two polytopes is contained in the other one, then the formulations are incomparable. In this case, there exist instances of FC-USPR such the first one yields a strictly better LP bound and there exist instances such that the latter one yield a strictly better bound.

First, consider the two polytopes  $\text{USPA}_z$  and  $\text{USPF}_z$  corresponding to the LP relaxations of (FC-ARCIP) and (FC-FWDIP), respectively. By Proposition (7.9), these two polytopes coincide.

**Theorem 7.24** *For all instances of FC-USPR, we have  $\text{USPA}_z = \text{USPF}_z$ .*



**Figure 7.3** Example for a fractional routing and arc capacities that are valid in the path variable based model but invalid in the arc flow variable based model.

**Proof.** Follows directly from Proposition (7.9).  $\square$

As already mentioned in Section 7.4, this implies that the LP relaxation of (FC-ARCIP) and that of (FC-FWDIP) always yield the same objective value.

Next, consider the two polytopes  $\text{USPA}_z$  and  $\text{USPR}_z$ , which correspond to the LP relaxations of (FC-ARCIP) and (FC-PATHIP), respectively. In the following, we show that neither  $\text{USPA}_z \subseteq \text{USPR}_z$  nor  $\text{USPR}_z \subseteq \text{USPA}_z$  holds in general.

**Theorem 7.25** *There exist instances of FC-USPR such that  $\text{USPA}_z \not\subseteq \text{USPR}_z$ .*

**Proof.** For the sake of brevity, we present an example where the routing paths of some commodities are already given and fixed, as they would be after some iterations in a branch and bound algorithm. Examples without pre-fixed routing paths can be constructed similarly.

Let  $D = (V, A)$  be the digraph shown in Figure 7.3, which consists of the 10 nodes

$$V := \{v_1, \dots, v_{10}\}$$

and the 14 arcs

$$A := \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5), (v_3, v_5), (v_6, v_7), (v_7, v_8), \\ (v_8, v_9), (v_9, v_{10}), (v_8, v_{10}), (v_1, v_7), (v_7, v_3), (v_6, v_2), (v_2, v_8)\}$$

with capacities  $u_a = 2$  for all  $a \in A$ . Within this digraph, we have the four commodities

$$K := \{(v_1, v_5), (v_1, v_{10}), (v_6, v_3), (v_6, v_8)\}$$

with demand values

$$d_{v_1, v_5} = d_{v_1, v_{10}} = 1 \quad \text{and} \quad d_{v_6, v_3} = d_{v_6, v_8} = 2.$$

We assume that also the routing paths of the two large commodities  $(v_6, v_3)$  and  $(v_6, v_8)$  are given and fixed. The prescribed routing paths are

$$\begin{aligned} P_{6,3} &:= (v_6, v_2, v_3) && \text{for commodity } (v_6, v_3) \text{ and} \\ P_{6,8} &:= (v_6, v_7, v_8) && \text{for commodity } (v_6, v_8). \end{aligned}$$

The polyhedra describing the feasible capacity variables in the corresponding restricted LP relaxations of (FC-PATHIP) and (FC-ARCIP) are

$$\begin{aligned} \text{USPR}_z^* &:= \text{conv} \{ z \in [0, 1]^A : \exists x \in [0, 1]^{\mathcal{P}(K)} \text{ such that} \\ &\quad (x, z) \text{ satisfies (7.2)–(7.4) and} \\ &\quad x_{P_{6,3}} = x_{P_{6,8}} = 1 \} , \text{ and} \\ \text{USPA}_z^* &:= \text{conv} \{ z \in [0, 1]^A : \exists y \in [0, 1]^{K \times A} \text{ such that} \\ &\quad (y, z) \text{ satisfies (7.6)–(7.8) and} \\ &\quad y_{(v_6, v_2)}^{(6,3)} = y_{(v_6, v_7)}^{(6,8)} = 1 \} . \end{aligned}$$

In order to show that  $\text{USPA}_z^* \subsetneq \text{USPR}_z^*$ , we consider the fractional capacity installation  $z \in \mathbb{R}_+^A$  defined as

$$z_a := \begin{cases} 1, & \text{if } a \in \{(v_6, v_2), (v_2, v_3), (v_6, v_7), (v_7, v_8)\}, \\ 1/2, & \text{if } a \in \{(v_1, v_2), (v_2, v_8), (v_1, v_7), (v_7, v_3)\}, \text{ and} \\ 1/4, & \text{otherwise.} \end{cases}$$

First, we show that  $z \notin \text{USPA}_z^*$ . For this, consider the fractional routing illustrated in Figure 7.3: The two commodities  $(v_6, v_3)$  and  $(v_6, v_8)$  are sent unsplit along their prescribed paths  $P_{6,3}$  and  $P_{6,8}$ , while commodity  $(v_1, v_5)$  is sent evenly split along the paths

$$P_{1,5}^A := (v_1, v_7, v_3, v_5) \quad \text{and} \quad P_{1,5}^B := (v_1, v_7, v_3, v_4, v_5)$$

and commodity  $(v_1, v_{10})$  is sent evenly split along the paths

$$P_{1,10}^A := (v_1, v_2, v_8, v_{10}) \quad \text{and} \quad P_{1,10}^B := (v_1, v_2, v_8, v_9, v_{10}).$$

As the capacities of the arcs  $(v_6, v_2)$ ,  $(v_2, v_3)$ ,  $(v_6, v_7)$ , and  $(v_7, v_8)$  are already saturated by routing the commodities  $(v_6, v_3)$  and  $(v_6, v_8)$  along the prescribed paths  $P_{6,3}$  and  $P_{6,8}$ , this is in fact the *only* fractional multicommodity flow routing that satisfies the capacities  $u_a z_a$  for all arcs  $a \in A$  and uses the prescribed paths  $P_{6,3}$  and  $P_{6,8}$ . With

$$\begin{aligned} R^1 &:= \{ ((v_6, v_3), (v_6, v_2)), ((v_6, v_3), (v_2, v_3)), \\ &\quad ((v_6, v_8), (v_6, v_7)), ((v_6, v_8), (v_7, v_8)), \\ &\quad ((v_1, v_5), (v_1, v_7)), ((v_1, v_5), (v_7, v_3)), \\ &\quad ((v_1, v_{10}), (v_1, v_2)), ((v_1, v_{10}), (v_2, v_8)) \} , \text{ and} \\ R^{0.5} &:= \{ ((v_1, v_5), (v_3, v_5)), ((v_1, v_5), (v_3, v_4)), ((v_1, v_5), (v_4, v_5)), \\ &\quad ((v_1, v_{10}), (v_8, v_{10})), ((v_1, v_{10}), (v_8, v_9)), ((v_1, v_{10}), (v_9, v_{10})) \} , \end{aligned}$$

the corresponding fractional arc routing  $y \in \mathbb{R}_+^{K \times A}$  is given by

$$y_a^{(s,t)} := \begin{cases} 1, & \text{for all } ((s,t), a) \in R^1, \\ 1/2, & \text{for all } ((s,t), a) \in R^{0,5}, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

Clearly,  $(y, z)$  satisfies (7.6) and (7.8). Hence, we have  $x \in \text{USPA}_z^*$  if and only if  $y$  satisfies all inequalities (7.7). As the forwarding

$$F := \{ (v_3, (v_2, v_3)), (v_8, (v_7, v_8)), (v_5, (v_7, v_3)), (v_{10}, (v_2, v_8)) \}$$

is an irreducible non-USPF in  $D$ , inequality (7.7) would be valid for the corresponding arc routing

$$R := \{ ((v_6, v_3), (v_2, v_3)), ((v_6, v_8), (v_7, v_8)), \\ ((v_1, v_5), (v_7, v_3)), ((v_1, v_{10}), (v_2, v_8)) \}.$$

However, inequality (7.7) is violated for this set  $R$  and, therefore,  $x \notin \text{USPA}_z^*$ .

To see that  $z \in \text{USPR}_z^*$ , consider the path routing vector  $x \in [0, 1]^{\mathcal{P}(K)}$  corresponding to the above fractional multicommodity flow routing, which is given by

$$x_P := \begin{cases} 1, & \text{for all } P \in \{P_{6,3}, P_{6,8}\}, \\ 1/2, & \text{for all } P \in \{P_{1,5}^A, P_{1,5}^B, P_{1,10}^A, P_{1,10}^B\}, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

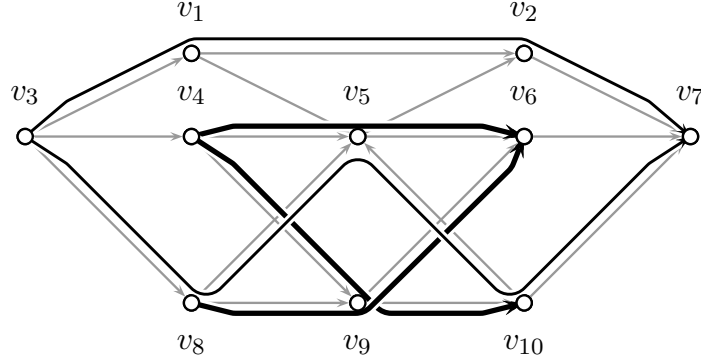
Clearly,  $(x, z)$  satisfies the inequalities (7.3) and (7.4). To see that  $x$  also satisfies all inequalities (7.2), recall that any irreducible non-USPS contains either 2 or at least 4 paths, see Theorem 5.7. As  $x_P \leq 1/2$  for all path but except  $P_{6,3}$  and  $P_{6,8}$ , (7.2) cannot be violated for a irreducible non-USPS  $S$  with  $|S| \geq 4$  and for no irreducible non-USPS  $S$  with  $|S| = 2$  and  $S \cap \{P_{6,3}, P_{6,8}\} = \emptyset$ . Hence, (7.2) can be violated only if  $F$  is a pair of paths from  $\{P_{6,3}, P_{6,8}, P_{1,5}^A, P_{1,5}^B, P_{1,10}^A, P_{1,10}^B\}$ . However, any such pair satisfies the Bellman property and thus forms an USPS. Consequently, all inequalities (7.2) are satisfied and  $z \in \text{USPR}_z^*$ .  $\square$

**Theorem 7.26** *There exist instances of FC-USPR such that  $\text{USPR}_z \not\subseteq \text{USPA}_z$ .*

**Proof.** Again, we present an example where the routing paths of some commodities are already given and fix.

Let  $D = (V, A)$  be the digraph illustrated in Figure 7.4. This digraph consists of the 10 nodes

$$V := \{v_1, \dots, v_{10}\}$$



**Figure 7.4** Example for a fractional routing and arc capacities that are valid in the arc flow variable based model but invalid in the path variable based model.

and the 17 arcs

$$A := \{ (v_3, v_1), (v_1, v_2), (v_1, v_5), (v_5, v_2), (v_2, v_7), (v_3, v_4), (v_4, v_5), \\ (v_5, v_6), (v_6, v_7), (v_3, v_8), (v_4, v_9), (v_5, v_{10}), (v_8, v_5), (v_9, v_6), \\ (v_{10}, v_7), (v_8, v_9), (v_9, v_{10}) \} ,$$

with arc capacities  $u_a = 1$  for all  $a \in A$ . Within this digraph, we are given the 4 commodities

$$K := \{ (v_4, v_6), (v_4, v_{10}), (v_8, v_6), (v_3, v_7) \}$$

with demand values  $d_{s,t} = 1$  for all  $(s, t) \in K$ .

Assume that the routing paths of the first 3 commodities  $(v_4, v_6)$ ,  $(v_4, v_{10})$ , and  $(v_8, v_6)$  are given and fixed as

$$\begin{aligned} P_{4,6} &:= (v_4, v_5, v_6) && \text{for commodity } (v_4, v_6), \\ P_{4,10} &:= (v_4, v_9, v_{10}) && \text{for commodity } (v_4, v_{10}), \text{ and} \\ P_{8,6} &:= (v_6, v_9, v_8) && \text{for commodity } (v_8, v_6). \end{aligned}$$

We denote the capacity polyhedra defined by the projections of the corresponding restricted LP relaxations by

$$\begin{aligned} \text{USPR}_z^* &:= \text{conv} \{ z \in [0, 1]^A : \exists x \in [0, 1]^{\mathcal{P}(K)} \text{ such that} \\ &\quad (x, z) \text{ satisfies (7.2)–(7.4) and} \\ &\quad x_{P_{4,6}} = x_{P_{4,10}} = x_{P_{8,6}} = 1 \} , \text{ and} \\ \text{USPA}_z^* &:= \text{conv} \{ z \in [0, 1]^A : \exists y \in [0, 1]^{K \times A} \text{ such that} \\ &\quad (y, z) \text{ satisfies (7.6)–(7.8) and} \\ &\quad y_{(v_4, v_5)}^{(4,6)} = y_{(v_4, v_9)}^{(4,10)} = y_{(v_8, v_9)}^{(8,6)} = 1 \} . \end{aligned}$$

We want to show that  $\text{USPR}_z^* \not\subseteq \text{USPA}_z^*$ . So, let  $z \in \mathbb{R}_+^A$  be defined as

$$z_a := \begin{cases} 1, & \text{for all } a \in P_{4,6} \cup P_{4,10} \cup P_{8,6}, \\ 0, & \text{for all } a \in \{(v_3, v_4), (v_1, v_5), (v_5, v_2), (v_6, v_7)\}, \text{ and} \\ 1/2, & \text{otherwise.} \end{cases}$$



First, we show that  $z \notin \text{USPR}_z^*$ . For this, consider the fractional routing where the first three commodities  $(v_4, v_6)$ ,  $(v_4, v_{10})$ , and  $(v_8, v_6)$  are routed unsplit along their respective prescribed paths and the fourth commodity  $(v_3, v_7)$  is routed evenly split along the two paths

$$P_{3,7}^A := (v_3, v_1, v_2, v_7) \quad \text{and} \quad P_{3,7}^B := (v_3, v_8, v_5, v_{10}, v_7) .$$

One easily observes that this routing is the *only* fractional multicommodity flow routing of the four commodities that satisfies the capacities  $u_{arc} z_a$  for all arcs  $a \in A$  and uses the prescribed paths for the first three commodities. Hence,  $z \in \text{USPR}_z^*$  if and only if  $x \in \mathbb{R}_+^{\mathcal{P}(K)}$  with

$$x_P := \begin{cases} 1, & \text{for all } P \in \{P_{4,6}, P_{4,10}, P_{8,6}\}, \\ 1/2, & \text{for all } P \in \{P_{3,7}^A, P_{3,7}^B\}, \text{ and} \\ 0, & \text{otherwise} \end{cases}$$

satisfies inequality (7.2) for all irreducible non-USPS  $S$  in  $D$ . However, the set  $S := \{P_{4,6}, P_{4,10}, P_{8,6}, P_{3,7}^B\}$  is an irreducible non-USPS and (7.2) is clearly violated for  $S$ . Thus,  $z \notin \text{USPR}_z^*$ .

On the other hand,  $z \in \text{USPA}_z^*$ . To see consider the arc routing  $y \in \mathbb{R}_+^{K \times A}$  defined by the above routing, which is given by

$$y_a^{(s,t)} := \begin{cases} 1, & \text{for all } ((s,t), a) \in R^1, \\ 1/2, & \text{for all } ((s,t), a) \in R^{0.5}, \text{ and} \\ 0, & \text{otherwise,} \end{cases}$$

where

$$\begin{aligned} R^1 &:= \{((s,t), a) : (s,t) \in \{(v_4, v_6), (v_4, v_{10}), (v_8, v_6)\} \text{ and} \\ &\quad a \in P_{4,6} \cup P_{4,10} \cup P_{8,6}\} , \text{ and} \\ R^{0.5} &:= \{((s,t), a) : (s,t) = (v_3, v_7) \text{ and } a \in P_{3,7}^A \cup P_{3,7}^B\} . \end{aligned}$$

Clearly,  $(y, z)$  satisfy all constraints (7.6) and (7.8). To see that  $y$  satisfies also all constraints (7.7), observe that the prescribed paths  $P_{4,6}$ ,  $P_{4,10}$ ,  $P_{8,6}$  form an USPS. Hence, if inequality (7.7) were violated for some arc routing  $R \subseteq K \times A$ , then  $R$  must contain some pair  $((s,t), a)$  with  $y_a^{(s,t)} > 0$  that does not correspond to the three prescribed paths. And this can only be one of the pairs  $((v_3, v_7), a)$  with  $a \in P_{3,7}^A \cup P_{3,7}^B$ . Since  $y_a^{(v_3, v_7)} = 1/2$  for all these pairs  $((v_3, v_7), a)$ ,  $R$  can contain only one of them for (7.7) to be violated. As, furthermore,  $R$  cannot contain a pair  $((s,t), a)$  with  $y_a^{(s,t)} = 0$  for (7.7) being violated,  $R$  must be a subset of one the sets  $R^{((s,t), a)} := R^1 \cup \{((s,t), a)\}$  with  $((s,t), a) \in R^{0.5}$ . These eight sets, however, are easily verified to be valid arc routings: Each arc routing where the three paths  $P_{4,6}$ ,  $P_{4,10}$  and  $P_{8,6}$  are entirely prescribed but only one routing arc in  $P_{3,7}^A$

or  $P_{3,7}^B$  is given for commodity  $(v_3, v_7)$  can be extended to a valid unsplittable shortest path routing for all four commodities. Consequently, none of the inequalities (7.7) can be violated by the arc routing  $y$ , which implies that  $z \in \text{USPA}_z^*$ .  $\square$

Theorems 7.25 and 7.26 show that neither  $\text{USPA}_z \subseteq \text{USPR}_z$  nor  $\text{USPR}_z \subseteq \text{USPA}_z$  holds in general. Hence, it is not clear a-priori which of the two formulations (FC-PATHIP) or (FC-ARCIP) yields the better LP bound. The LP relaxations of these two formulations are incomparable in general.

With Proposition 7.24 it follows immediately that also the LP relaxations of (FC-PATHIP) and of (FC-FWDIP) are incomparable.

**Remark 7.27** Theorems 7.25 and 7.26 carry over straightforward to the corresponding LP relaxations of (CON-PATHIP) and (CON-ARCIP). They also hold for restricted LP relaxations of (CAP-PATHIP) and (CAP-ARCIP) in which the routing paths for some commodities or some of the capacity variables are fixed. The unrestricted LP relaxations of (CAP-PATHIP) and (CAP-ARCIP) without any prescribed routing paths or capacities, however, always yield the same optimal solution value.

In the following chapter we show how to strengthen the LP relaxations of the basic path routing and arc routing formulations discussed above.

## Chapter 8

# Valid Inequalities

In this chapter, we derive several classes of valid inequalities for the polyhedra associated with the integer programming models introduced in Chapter 7. Our main objective is to relate the facial structure of these integer programming polyhedra to that of other well-known and well-studied polyhedra and to illustrate how valid inequalities for the latter ones carry over to valid and sometimes even facet-defining inequalities for the former ones.

There exists a vast literature on valid inequalities for various related network design and routing problems. Many valid and facet-defining inequalities have been derived for network connectivity problems [25, 58, 59, 98, 106, 107, 102, 104, 135], unsplittable or fractional multicommodity flow problems [4, 6, 8, 14, 15, 31, 30, 55, 68, 66], or special link capacity structures [1, 43, 109, 136, 188]. Because unsplittable shortest path routing is more restrictive than the routing schemes assumed in these articles, the inequalities carry over immediately to the polyhedra associated with the corresponding unsplittable shortest path routing problems. Therefore, we focus on two substructures that arise only in unsplittable shortest path routing polyhedra and on one technique that was originally proposed for unsplittable flow polyhedra in this chapter.

In Section 8.1, we discuss how the facets of the considered unsplittable shortest path routing polyhedra are related to the facets of the independence system polyhedra associated with the family of unique shortest path routings and unique shortest path forwardings. In Section 8.2, we review an approach proposed by Belaidouni and Ben-Ameur [20] for deriving strong metric-type inequalities for unsplittable flow routing polyhedra. The so-called super-additive metric inequalities obtained with this approach are stronger than those that can be derived by strengthening the classical metric inequalities for fractional multicommodity flow routing. In Section 8.3 we finally illustrate how valid inequalities for the unsplittable shortest path polyhedra can be obtained from the precedence constrained knapsack polytope that is defined by single arc capacity constraint and all binary routing variables involving this arc.

Throughout this chapter, we denote the polyhedra associated with the integer linear programming models introduced above as follows:

$$\begin{aligned}
\text{USPR}^{\text{FC}}(K, u) &:= \text{conv} \left\{ (x, z) \in \{0, 1\}^{\mathcal{P}(K) \cup A} : \right. \\
&\quad \left. (x, z) \text{ satisfies (7.2)–(7.4)} \right\} \\
\text{USPR}^{\text{Cap}}(K, u) &:= \text{conv} \left\{ (x, z) \in \{0, 1\}^{\mathcal{P}(K)} \times \mathbb{Z}_+^A : \right. \\
&\quad \left. (x, z) \text{ satisfies (7.2)–(7.4)} \right\} \\
\text{USPR}^{\text{Con}}(K, u) &:= \text{conv} \left\{ (x, L) \in \{0, 1\}^{\mathcal{P}(K)} \times \mathbb{R}_+ : \right. \\
&\quad \left. (x, L) \text{ satisfies (7.2), (7.3), and (7.5)} \right\} \\
\text{USPA}^{\text{FC}}(K, u) &:= \text{conv} \left\{ (y, z) \in \{0, 1\}^{(V \times A) \cup A} : \right. \\
&\quad \left. (y, z) \text{ satisfies (7.6)–(7.8)} \right\} \\
\text{USPA}^{\text{Cap}}(K, u) &:= \text{conv} \left\{ (y, z) \in \{0, 1\}^{V \times A} \times \mathbb{Z}_+^A : \right. \\
&\quad \left. (y, z) \text{ satisfies (7.6)–(7.8)} \right\} \\
\text{USPA}^{\text{Con}}(K, u) &:= \text{conv} \left\{ (y, L) \in \{0, 1\}^{V \times A} \times \mathbb{R}_+ : \right. \\
&\quad \left. (y, L) \text{ satisfies (7.6), (7.7), and (7.9)} \right\}
\end{aligned}$$

## 8.1 Routing Inequalities

In this section we show how to derive valid and facet-defining inequalities for unsplittable shortest path routing polyhedra from the fact that the unique shortest path systems (USPS) and the unique shortest path forwardings (USPF) in a digraph form independence systems. Applying known results and techniques for general independence system polytopes, one immediately obtains valid and facet-defining inequalities for  $\text{USPR}$  and  $\text{USPA}$ . These inequalities then carry over trivially to the polyhedra  $\text{USPR}^{\text{FC}}$ ,  $\text{USPR}^{\text{Cap}}$ ,  $\text{USPR}^{\text{Con}}$ ,  $\text{USPA}^{\text{FC}}$ ,  $\text{USPA}^{\text{Cap}}$ , and  $\text{USPA}^{\text{Con}}$  associated with our integer linear programming formulations. For the arc routing formulation we present some additional inequalities, which cannot be derived directly from the independence system characterization.

Note that the inequalities discussed in this section are independent of the given capacities  $u_a$ ,  $a \in A$ . They only reflect the fact that the routing must be an unsplittable shortest path routing.

### 8.1.1 Path Routing Formulation

First, we examine the polyhedra associated with the path routing formulations. Let  $\mathcal{I}_{\text{USPS}} \subseteq 2^{\mathcal{P}}$  be the independence system of unique shortest path systems and consider its associated **independence system polytope**

$$\begin{aligned}
\text{IND}(\mathcal{I}_{\text{USPS}}) &:= \text{conv} \left\{ \chi(S) : S \subseteq \mathcal{P}, S \text{ is an USPS} \right\} \\
&= \text{conv} \left\{ x \in \{0, 1\}^{\mathcal{P}} : x \text{ satisfies (7.2)} \right\}.
\end{aligned}$$

This polytope is equivalent to the stable set polytope  $\text{STAB}(\mathcal{H}_{USPS})$  of the corresponding conflict hypergraph  $\mathcal{H}_{USPS} = (\mathcal{P}, \mathcal{C}_{USPS})$ , whose nodes are the paths in  $D$  and whose hyperedges are the irreducible non-USPSs (i.e., the circuits  $\mathcal{C}_{USPS}$  of the independence system  $\mathcal{I}_{USPS}$ ).

It is easy to see that the unsplittable shortest path routing polytope  $\text{USPR}$  is a face of  $\text{IND}(\mathcal{I}_{USPS})$ . In fact,

$$\text{USPR} = \text{conv} \left\{ x \in \text{IND}(\mathcal{I}_{USPS}) : x_P = 0 \quad \forall P \in \mathcal{P} \setminus \mathcal{P}(K) \text{ and } \sum_{P \in \mathcal{P}(s,t)} x_P = 1 \quad \forall (s,t) \in K \right\}.$$

Therefore, any valid inequality for  $\text{IND}(\mathcal{I}_{USPS})$  is valid for  $\text{USPR}$  too and, in consequence, also for the polyhedra  $\text{USPR}^{\text{FC}}$ ,  $\text{USPR}^{\text{Cap}}$ ,  $\text{USPR}^{\text{Con}}$ . More precisely, we have the following two results.

**Observation 8.1**

- (i) *Each valid inequality for  $\text{IND}(\mathcal{I}_{USPS})$  is also valid for  $\text{USPR}$ .*
- (ii) *Each facet of  $\text{USPR}$  is supported by an inequality  $\alpha^T x \leq \alpha_0$  that defines a facet of  $\text{IND}(\mathcal{I}_{USPS})$ .*

**Observation 8.2**

- (i) *Each valid inequality for  $\text{USPR}$  is also valid for  $\text{USPR}^{\text{FC}}$ .*
- (ii) *Each valid inequality for  $\text{USPR}$  is also valid for  $\text{USPR}^{\text{Cap}}$ . Furthermore, each facet-defining inequality for  $\text{USPR}$  is facet-defining for  $\text{USPR}^{\text{Cap}}$  too.*
- (iii) *Each valid inequality for  $\text{USPR}$  is also valid for  $\text{USPR}^{\text{Con}}$ . If*

$$u_a \geq \max \{ f_a(\lambda) : \lambda \text{ is a valid metric for } K \} \quad \text{for all } a \in A,$$

*then every facet-defining inequality for  $\text{USPR}$  is also facet-defining for  $\text{USPR}^{\text{Con}}$ .*

Note that Observation 8.1 not only implies that all valid inequalities of  $\text{IND}(\mathcal{I}_{USPS})$  are valid for  $\text{USPR}$ . It also states that it is sufficient to study the facets of  $\text{IND}(\mathcal{I}_{USPS})$  in order to obtain a complete facial description of  $\text{USPR}$ . By Observation 8.2, the corresponding inequalities then are also valid or even facet defining for the polyhedra associated with our integer linear programming formulations.

There is a vast literature related to the properties and to valid inequalities for independence system polytopes, stable set polytopes, or the closely related set covering polytope, see for example Edmonds [76], Padberg [151], Nemhauser and Trotter [145], Hammer et al. [111], Euler et al. [82], Conforti

and Laurant [61], Cornuéjols and Sassano [62], Laurent [131], Sassano [174], and Müller and Schulz [144] to mention just a few. Applying this machinery to the independence system polytope  $\text{IND}(\mathcal{I}_{USPS})$  yields numerous classes of valid and in some cases even facet-defining inequalities for USPR.

In the following, we review only the very basic properties of  $\text{IND}(\mathcal{I}_{USPS})$  and USPR and present the class of general rank inequalities and one of its subclasses, the clique inequalities.

A standard result of Hammer et al. [111] on the facial structure of general independence systems implies that the following for  $\text{IND}(\mathcal{I}_{USPS})$ .

**Theorem 8.3 (Hammer et al. [111])** *Each facet of  $\text{IND}(\mathcal{I}_{USPS})$  is either*

- (i) *a trivial facet induced by an inequality  $x_P \geq 0$ ,  $P \in \mathcal{P}$ ,*
- (ii) *or induced by a unique inequality  $\alpha^T x \leq 1$ . Moreover,  $\alpha_P \geq 0$  for all  $P \in \mathcal{P}$ .*

With Observation 8.1, one obtains the following properties for the facets of USPR.

**Corollary 8.4** *Each facet of USPR is either*

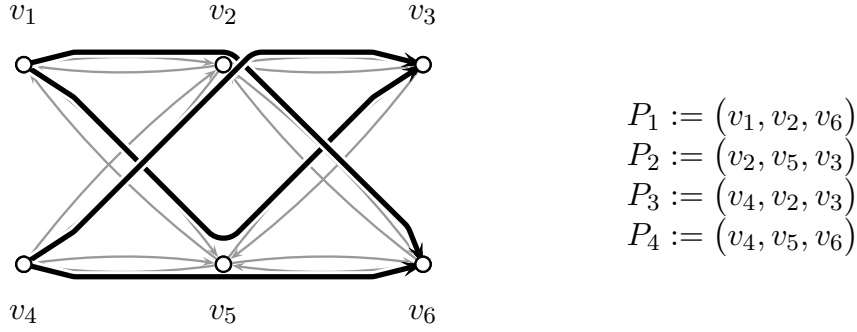
- (i) *a trivial facet induced by an inequality  $x_P \geq 0$ ,  $P \in \mathcal{P}(K)$ ,*
- (ii) *or induced by some inequality  $\alpha^T x \leq 1$  with  $\alpha_P \geq 0$  for all  $P \in \mathcal{P}(K)$ .*

One class of non-trivial facets of  $\text{IND}(\mathcal{I}_{USPF})$  and USPR is induced by the so-called rank inequalities. Let  $r : 2^{\mathcal{P}} \rightarrow \mathbb{Z}_+$  be the rank function of the independence system  $\mathcal{I}_{USPS}$ , i.e., for each path set  $S \subseteq \mathcal{P}$  the number  $r(S)$  is the maximum cardinality  $|R|$  of a unique shortest path system  $R \in \mathcal{I}_{USPS}$  with  $R \subseteq S$ . For any path set  $S \subseteq \mathcal{P}(K)$ , the corresponding **rank inequality** is

$$\sum_{P \in S} x_P \leq r(S) . \quad (8.1)$$

It is well known that rank inequalities are valid for general independence system polytopes, see Edmonds [76] or Hammer et al. [111] for example. With Observation 8.1 we thus obtain the following.

**Theorem 8.5** *For any path set  $S \subseteq \mathcal{P}(K)$ , the corresponding rank inequality (8.1) is valid for USPR.*



**Figure 8.1** Non-closed irreducible non-USPS.

In general, the separation problem for rank inequalities (8.1) is  $\mathcal{NP}$ -complete.

The USPS circuit inequalities (7.2) used in our path routing formulation are a special case of rank inequalities, given by the sets  $S \subseteq \mathcal{P}(K)$  with  $r(S) = |S| - 1$ . The following example shows that these inequalities are not necessarily facet-defining for  $\text{IND}(\mathcal{I}_{USPS})$  or USPR.

**Example 8.1** The path system  $S$  shown in Figure 8.1 is an irreducible non-USPS with  $r(S) = |S| - 1 = 3$ . Now consider the path  $P_0 := (v_1, v_2, v_3, v_5, v_4)$ . Since this path conflicts with all other paths  $P_1, \dots, P_4$ , we have  $r(S) = r(S \cup \{P_0\}) = 3$ . Hence, the rank inequality  $\sum_{i=1}^4 x_{P_i} \leq 3$  for  $S$  is dominated by the rank inequality  $\sum_{i=0}^4 x_{P_i} \leq 3$  for  $S \cup \{P_0\}$  and, consequently, the first one can not be facet defining for  $\text{IND}(\mathcal{I}_{USPS})$ .  $\triangle$

Various sub-classes of rank inequalities (8.1) have been studied in the literature. Most of them are named according to the substructures their support induces in the corresponding conflict hypergraph. Often they generalize well known inequalities for stable set polyhedra of simple graphs. Examples are the generalized clique inequalities, generalized odd cycle or anti-cycle inequalities, or generalizations of anti-web inequalities, see Nemhauser and Trotter [145], Euler et al. [82], Laurent [131], and Müller and Schulz [144]. We will not reformulate these inequalities or the conditions under which each of these inequalities is facet defining here. In general, the supporting substructures of these inequalities and the conditions for being facet-defining have no natural interpretation in terms of path sets. We only want to mention the class of simple clique inequalities, because these inequalities are heavily used in our solution algorithm.

Given a set  $S \subseteq \mathcal{P}$  of pairwise conflicting paths, the corresponding **(USPS) clique inequality** is

$$\sum_{P \in S} x_P \leq 1 \quad (= r(S)) . \quad (8.2)$$

It follows directly from Theorem 8.5 and Observation 8.2 that the clique inequality is valid for USPR.

**Corollary 8.6** *For any set  $S \subseteq \mathcal{P}(K)$  of pairwise conflicting paths, the corresponding clique inequality (8.2) is valid for USPR.*

One easily verifies that the clique inequality (8.2) is facet-defining for the independence system polytope  $\text{IND}(\mathcal{I}_{USPS})$  if and only if  $S$  is maximal with respect to inclusion, i.e., if there is no path  $P \in \mathcal{P}(K) \setminus S$  that conflicts with all paths in  $S$ .

Clique inequalities are of special interest in many applications. Often they can be separated very efficiently by combinatorial heuristics, even though the exact separation problem for (maximum) clique inequalities is  $\mathcal{NP}$ -hard.

Together with the demand constraints 7.3, the clique inequalities (8.2) imply the so-called **subpath inequalities** (also called precedence or ordering constraints)

$$x_Q \geq x_P \quad \forall Q \subset P. \quad (8.3)$$

One easily verifies that, for each path pair  $Q \subset P$ , the subpath constraint (8.3) is a linear combination of the (not necessarily maximal) clique inequality

$$\sum_{P' \in \mathcal{P}(s_Q, t_Q) \setminus \{Q\}} x_{P'} + x_P \leq 1$$

and the demand constraint

$$\sum_{P' \in \mathcal{P}(s_Q, t_Q)} x_{P'} = 1.$$

Hence, the subpath constraints (8.3) are valid for USPR.

### 8.1.2 Arc Routing Formulation

Rank and clique inequalities for the arc routing formulation are derived (in principle) analogous to those for the path routing formulation.

Let  $\mathcal{I}_{USPF} \subseteq 2^{V \times A}$  be the independence system of unique shortest path forwardings and consider its associated independence system polytope

$$\begin{aligned} \text{IND}(\mathcal{I}_{USPF}) := & \text{conv} \{ \chi(F) : F \subseteq V \times A, F \text{ is an USPF} \} \\ & \text{conv} \{ x \in \{0, 1\}^{V \times A} : x \text{ satisfies (7.7a)} \}. \end{aligned}$$

This polytope is equivalent to the stable set polytope  $\text{STAB}(\mathcal{H}_{USPF})$  on the corresponding circuit hypergraph  $\mathcal{H}_{USPF} := (V \times A, \mathcal{C}_{USPF})$  again.

On the other hand, consider the polytope that is given by projecting the unsplittable shortest path forwarding polytope USPF onto the space of forwarding variables, i.e.,

$$\begin{aligned} \text{USPF}_x := & \text{conv} \{ x \in \{0, 1\}^{V \times A} : \exists y \in \mathbb{R}^{K \times A} \text{ with } (y, x) \in \text{USPF} \} \\ = & \text{conv} \{ x \in \{0, 1\}^{V \times A} : \exists y \in \mathbb{R}^{K \times A} \text{ with } y \in \text{USPA s.t.} \\ & (y, x) \text{ satisfies (7.7a) and (7.7b)} \}. \end{aligned}$$



It is obvious that  $\text{USPF}_x \subseteq \text{IND}(\mathcal{I}_{\text{USPF}})$ . Therefore, any valid inequality for  $\text{IND}(\mathcal{I}_{\text{USPF}})$  is valid also for  $\text{USPF}_x$  and  $\text{USPF}$ . Applying Theorem 7.11, any valid inequality for  $\text{USPF}$  with only non-negative coefficients then translates into a set of valid inequalities for  $\text{USPA}$ . Since all non-trivial facets of  $\text{IND}(\mathcal{I}_{\text{USPF}})$  are supported by inequalities with only non-negative coefficients [111], every non-trivial facet of  $\text{IND}(\mathcal{I}_{\text{USPF}})$  yields a set of valid inequalities for  $\text{USPA}$ .

Various classes of valid inequalities for  $\text{USPA}$  thus can be derived by applying the results for general independence system polytopes known from the literature to  $\text{IND}(\mathcal{I}_{\text{USPF}})$  and reformulating the inequalities in terms of arc routing variables. In the following, we review only the most basic ones.

Let  $r : 2^V \times A \rightarrow \mathbb{Z}_+$  be the rank function of  $\mathcal{I}_{\text{USPF}}$ .

**Theorem 8.7** *For any forwarding  $F \subseteq V \times A$ , the corresponding (**USPF**) rank inequality*

$$\sum_{(t,a) \in F} x_a^t \leq r(F) \quad (8.4)$$

*is valid for  $\text{USPF}_x$  and  $\text{USPF}$ .*

The  $\text{USPF}$  circuit inequalities (7.7) and (7.7a) used in the two arc routing formulations of unsplittable shortest path routings are the subset of rank inequalities corresponding to the circuits of  $\mathcal{I}_{\text{USPF}}$ . It is not difficult to see that the circuit inequalities are not necessarily facet-defining for  $\text{USPA}$  or  $\text{USPF}$ . A counter-example can be constructed analogous to Example 8.1.1.

Given a set  $F \subseteq V \times A$  of pairwise conflicting forwarding arcs, the corresponding (**USPF**) **clique inequality** is

$$\sum_{(t,a) \in F} x_a^t \leq 1 \quad (= r(F)) . \quad (8.5)$$

It is easy to verify that the clique inequality is valid for  $\text{USPF}_x$  and  $\text{USPF}$ . A subset of the  $\text{USPF}$  clique inequalities are the **out-degree inequalities**

$$\sum_{a \in \delta^+(v)} x_a^t \leq 1 \quad \forall t, v \in V . \quad (8.6)$$

With Theorem 7.11, each out-degree inequality (8.6) carries over to the set of inequalities

$$\sum_{a \in \delta^+(v)} y_a^{s(a),t} \leq 1 \quad \forall v, t \in V, s : \delta^+(v) \rightarrow V \text{ with } (s(a), t) \in K. \quad (8.7)$$

By Theorem 7.11, these inequalities are valid for  $\text{USPA}$ .

We shall point out that independence system based inequalities can be derived also for the ‘reverse’ problem. By considering also the independence system of unique shortest path ‘anti-forwardings’, which is equivalent to the system of unique shortest path forwarding in the problem instance where all arcs and all commodities are reversed, one obtains a set of reverse valid inequalities for the arc routing polytope USPA.

The corresponding reverse USPF clique inequalities contain the set of **in-degree inequalities** which, after applying the corresponding version of Theorem 7.11, lead to the following inequalities in terms of arc routing variables:

$$\sum_{a \in \delta^-(v)} y_a^{s,t(a)} \leq 1 \quad \forall v, s \in V, t : \delta^-(v) \rightarrow V \text{ with } (s, t(a)) \in K \quad (8.8)$$

Clearly, these inequalities are valid for USPA.

Finally, we will present some additional valid inequalities for USPA, which are not derived from the independence system polytopes of forwardings or anti-forwardings. These simple inequalities operate directly on the arc routing variables. They all exploit the fact that a commodity  $(s, t)$  whose terminals  $s$  and  $t$  are both contained in the routing path  $P$  for another commodity must be routed on the corresponding subpath  $P[s, t]$  of the path  $P$ .

**Proposition 8.8** *For each pair of commodities  $(s, t), (s, v) \in K$  and for each arc  $a \in A$ , the corresponding **(source) subpath inequality***

$$y_a^{s,v} - y_a^{s,t} + \sum_{e \in \delta^-(v)} y_e^{s,t} \leq 1 \quad (8.9)$$

*is valid for USPA.*

**Proof.** Let  $(s, t), (s, v) \in K$  and consider an arbitrary vertex  $y \in \{0, 1\}^{K \times A}$  of USPA. Let  $P$  and  $Q$  be the two path defined by  $y$  for the commodities  $(s, t)$  and  $(s, v)$  respectively.

Clearly, we either have  $\sum_{e \in \delta^-(v)} y_e^{s,t} = 0$  or  $\sum_{e \in \delta^-(v)} y_e^{s,t} = 1$ .

If  $\sum_{e \in \delta^-(v)} y_e^{s,t} = 0$ , inequality (8.9) reduces to  $y_a^{s,v} - y_a^{s,t} \leq 1$  for all arcs  $a \in A$ . This is satisfied trivially, because  $y \in \{0, 1\}^{K \times A}$ .

So, assume  $\sum_{e \in \delta^-(v)} y_e^{s,t} = 1$ . This implies that  $v$  is contained in  $P$ . As  $P$  and  $Q$  also share the common source  $s$ , the Bellman property implies that  $Q \subset P$ . Consequently, we have  $y_a^{s,v} \leq y_a^{s,t}$  for all arc  $a \in A$  and inequality (8.9) is satisfied.  $\square$

An analogous inequality can be formulated by considering commodities that share a common destination.

**Proposition 8.9** *For each pair of commodities  $(s, t), (v, t) \in K$  and for each arc  $a \in A$ , the corresponding **(destination) subpath inequality***

$$y_a^{v,t} - y_a^{s,t} + \sum_{e \in \delta^-(v)} y_e^{s,t} \leq 1 . \quad (8.10)$$

*is valid for USPA.*

**Proof.** Analogous to Proposition 8.8. □

Inequalities for triples of commodities can be formulated similarly.

**Proposition 8.10** *For each triple of commodities  $(s, v), (v, t), (s, t) \in K$  and for each arc  $a \in A$ , the corresponding **(triple) subpath inequality***

$$y_a^{s,v} - y_a^{s,t} + \sum_{e \in \delta^-(v)} y_e^{s,t} \leq 1 . \quad (8.11)$$

*is valid for USPA.*

**Proof.** Analogous to Proposition 8.8. □

Further inequalities of this type can be formulated easily.

In general, none of the subpath inequalities (8.9), (8.10), and (8.11) is facet-defining for USPA. Yet, they proved to be very useful in practice.

## 8.2 Superadditive Metric Inequalities

In this section, we review a class of inequalities that can be obtained by strengthening the well-known metric inequalities derived by Iri [117] and Onaga and Kakusho [147] for fractional multicommodity flows. The presented approach was proposed by Belaidouni and Ben-Ameur [20] for unsplittable flow problems. The strengthening technique is based on applying a non-decreasing superadditive function to the capacity constraints before aggregating them according to some given metric.

Throughout this section, let  $D = (V, A)$  be a digraph with arc capacities  $u_a$ ,  $a \in A$ , and  $K \subseteq V \times V$  be a set of commodities with demand values  $d_{s,t} \in \mathbb{Z}_+$ ,  $(s, t) \in K$ .

### 8.2.1 Classical Metric Inequalities

Many classes of valid inequalities for network design and routing problems have been derived from the well-known metric inequalities introduced by Iri [117] and Onaga and Kakusho [147].

**Theorem 8.11 (Iri [117], Onaga and Kakusho [147])** *Let  $u \in \mathbb{R}_+^A$  be a vector of arc capacities. There exists a fractional multicommodity flow routing of the commodities  $K$  that satisfies these arc capacities if and only if the **metric inequality***

$$\sum_{a \in A} \mu_a u_a \geq \sum_{(s,t) \in K} \pi_{s,t}^\mu d_{s,t} \quad (8.12)$$

holds for all  $\mu = (\mu_a)_{a \in A} \in \mathbb{R}_+^A$ , where  $\pi_{s,t}^\mu$  is the length of a shortest  $(s, t)$ -path with respect to  $\mu$ .

Theorem 8.11 follows straightforward by applying Farkas' lemma to the linear system

$$\begin{aligned} \sum_{P \in \mathcal{P}(s,t)} x_P &= 1 & \forall (s,t) \in K, \\ \sum_{P \in \mathcal{P}(K): a \in P} d_{s_P, t_P} x_P &\leq u_a & \forall a \in A, \\ x_P &\geq 0 & \forall P \in \mathcal{P}(K), \end{aligned}$$

which describes all fractional multicommodity flow routings  $x \in \mathbb{R}_+^{\mathcal{P}(K)}$  that satisfy the given arc capacities  $u_a$ ,  $a \in A$ .

Two important subclasses of the metric inequalities (8.12) are the graph partition inequalities and the directed cut inequalities. Let  $U_1 \dot{\cup} \dots \dot{\cup} U_k = V$  be a partition of the nodes. The corresponding *graph partition inequality* is the metric inequality defined by metric  $\mu = \chi(\delta(U_1, \dots, U_k))$ , i.e.,

$$\mu_a := \begin{cases} 1, & \text{if } a = (u, v) \text{ with } u \in U_i, v \in U_j, i \neq j, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

For  $k = 2$ , the partition inequality is also called (*undirected*) *cut inequality*. The *directed cut inequality* corresponding a node set  $U \subset V$  is given by the metric  $\mu = \chi(\delta^+(U))$ , i.e.,

$$\mu_a := \begin{cases} 1, & \text{if } a = (u, v) \text{ with } u \in U, v \in V \setminus U, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

There are numerous publications related to the strength of these inequalities in general or related to special cases where the cut or graph partition inequalities sufficiently characterize the existence of a fractional multicommodity flow routing for some given capacities. Overviews of such results can be found in the surveys by Frank [94, 95] and Schrijver [176], for example. The most famous results in this context certainly are the Max-Flow-Min-Cut Theorem of Ford and Fulkerson [91] and Hu's Theorem [116]. The

Max-Flow-Min-Cut Theorem implies that the (directed) cut inequalities sufficiently characterize the existence of a fractional single commodity routing in a (directed) graph, while Hu's Theorem shows that the undirected cut inequalities are also sufficient to describe fractional two-commodity routings in the undirected case.

Now consider the polytope

$$\text{MCF}^{\text{FC}}(K, u) := \text{conv} \left\{ (x, z) \in \mathbb{R}_+^{\mathcal{P}(K)} \times \{0, 1\}^A : \right. \\ \left. (x, z) \text{ satisfies (7.3) and (7.4) } \right\},$$

which corresponds to the fixed charge network design problem with fractional multicommodity flow routing. With  $u_a := u_a z_a$ , it follows immediately from Theorem 8.11 that, for each metric  $\mu \in \mathbb{R}_+^A$ , the corresponding metric inequality

$$\sum_{a \in A} (\mu_a u_a) z_a \geq \sum_{(s,t) \in K} \pi_{s,t}^\mu d_{s,t} \quad (8.13)$$

is valid for  $\text{MCF}^{\text{FC}}$ . Note that this inequality involves only the binary capacity variables  $z_a$ ,  $a \in A$ . So, for any metric  $\mu \in \mathbb{R}_+^A$ , it defines a 0/1-covering (or, equivalently, a 0/1-knapsack) problem on these variables. Clearly, any inequality that is valid for the corresponding 0/1-covering polytope

$$\text{ICOV}(\mu) := \text{conv} \left\{ z_a \in \{0, 1\}^A : z \text{ satisfies (8.13) } \right\}$$

is also valid for  $\text{MCF}^{\text{FC}}$ .

Using basically this observation, several classes of valid and facet defining inequalities for  $\text{MCF}^{\text{FC}}$  and for more general multicommodity flow network design polytopes have been derived in the literature. [15] approached network design problems with basic *strengthened cut inequalities*. Dahl and Stoer [66, 67] introduced the so-called *band and partition band inequalities*, which correspond to cover inequalities for the knapsacks defined by (8.13) for arbitrary metrics and for graph partition metrics. Alevras et al. [5] and Wessälly [190] generalized these inequalities to *2-band and strengthened 2-band inequalities*, Alevras et al. also used so-called *strengthened metric inequalities*, which can be obtained by applying standard integer routing techniques to (8.13). Further classes of strong valid inequalities for network design problems with multicommodity flow routing have been discussed by Bienstock et al. [30, 31].

Since  $\text{USPR}^{\text{FC}} \subseteq \text{MCF}^{\text{FC}}$ , these inequalities are valid of  $\text{USPR}^{\text{FC}}$  as well.

### 8.2.2 Superadditive Metric Inequalities

For unsplittable flow routing, stronger inequalities can be obtained by applying a non-decreasing superadditive function to the coefficients of the capacity constraints (7.4) before aggregating them to a metric-like inequality.

**Definition 8.12**

- (i) A function  $\Gamma : \mathbb{R}^m \rightarrow \mathbb{R}$  is called **superadditive** if  $\Gamma(a) + \Gamma(b) \leq \Gamma(a + b)$  for all  $a, b \in \mathbb{R}^m$ .
- (ii) A function  $\Gamma : \mathbb{R}^m \rightarrow \mathbb{R}$  is called **non-decreasing** if  $\Gamma(a) \leq \Gamma(b)$  for all  $a \leq b$ .

It is well known that non-decreasing superadditive functions can be used to generate valid inequalities for integer programming polyhedra, see Nemhauser and Wolsey [146] or Wolsey [194] for example.

Consider the polyhedron  $P = \text{conv} \{x \in \mathbb{Z}_+^n : Ax \leq b\}$  defined by a matrix  $A \in \mathbb{R}^{m \times n}$  and a vector  $b \in \mathbb{R}^m$ , and denote the  $i$ -th column of  $A$  by  $a_i$  for each  $i = 1, \dots, n$ . If  $\Gamma : \mathbb{R}^m \rightarrow \mathbb{R}$  is a non-decreasing and superadditive function, then the inequality

$$\sum_{i=1}^n \Gamma(a_i) x_i \leq \Gamma(b) \quad (8.14)$$

is valid for  $P$ . Even more, each facet of  $P$  can be derived from  $A$  and  $b$  by applying an appropriate non-decreasing superadditive function.

**Theorem 8.13 (Burdet and Johnson [47], Jeroslow [119])** *Every valid inequality for a non-empty polyhedron  $P = \text{conv} \{x \in \mathbb{Z}_+^n : Ax \leq b\}$  is equivalent to or dominated by an inequality of type (8.14).*

It follows immediately from (8.14) that, if  $\sum_i \alpha_i x_i \leq \beta$  is a valid inequality for  $P$  and  $\Gamma : \mathbb{R} \rightarrow \mathbb{R}$  is a one-dimensional non-decreasing superadditive function, then also  $\sum_i \Gamma(\alpha_i) x_i \leq \Gamma(\beta)$  is valid for  $P$ .

Applying this observation to the polytope

$$\text{UFP}^{\text{FC}}(K, u) := \text{conv} \left\{ (x, z) \in \{0, 1\}^{\mathcal{P}(K) \cup A} : \right. \\ \left. (x, z) \text{ satisfies (7.3) and (7.4)} \right\},$$

which corresponds to the fixed charge network design problem with unsplittable flow routing, Belaidouni and Ben-Ameur obtained the following result.

**Theorem 8.14 (Belaidouni and Ben-Ameur [20])** *For any non-negative metric  $\mu \in \mathbb{R}_+^A$  and any non-decreasing superadditive function  $\Gamma : \mathbb{R} \rightarrow \mathbb{R}$ , the **superadditive metric inequality***

$$\sum_{a \in A} (\mu_a \Gamma(u_a)) z_a \geq \sum_{(s,t) \in K} \pi_{s,t}^\mu \Gamma(d_{s,t}) \quad (8.15)$$

is valid for  $\text{UFP}^{\text{FC}}$ , where  $\pi_{s,t}^\mu$  is the length of a shortest  $(s, t)$ -path with respect to  $\mu$ .

**Proof.** Let  $\mu \in \mathbb{R}_+^A$  be a non-negative metric and  $\Gamma : \mathbb{R} \rightarrow \mathbb{R}$  be a non-decreasing superadditive function.

By the definition of  $\text{UFP}^{\text{FC}}$ , all capacity inequalities (7.4) are valid for  $\text{UFP}^{\text{FC}}$ . Because  $z_a \in \{0, 1\}$  for each  $a \in A$ , also each strengthened capacity constraints

$$\sum_{P \in \mathcal{P}(K): a \in P} \Gamma(d_{s_P, t_P}) \cdot x_P \leq \Gamma(u_a) \cdot z_a \quad \forall a \in A \quad (8.16)$$

are valid for  $\text{UFP}^{\text{FC}}$ .

Naturally, any non-negative linear combination of these inequalities is valid for  $\text{UFP}^{\text{FC}}$  too. Hence, also the aggregated inequality

$$\begin{aligned} \sum_{a \in A} \mu_a \cdot \left( \sum_{P \in \mathcal{P}(K): a \in P} \Gamma(d_{s_P, t_P}) \cdot x_P \right) = \\ \sum_{P \in \mathcal{P}(K)} \Gamma(d_{s_P, t_P}) \cdot \left( x_P \sum_{a \in P} \mu_a \right) \leq \sum_{a \in A} \mu_a \Gamma(u_a) \cdot z_a \end{aligned}$$

is valid for  $\text{UFP}^{\text{FC}}$ . Since  $\sum_{P \in \mathcal{P}(K)} x_P = 1$  for all  $x \in \text{UFP}^{\text{FC}}$ , it follows that

$$\sum_{P \in \mathcal{P}(K)} \Gamma(d_{s_P, t_P}) \pi_{s,t}^\mu \leq \sum_{a \in A} \mu_a \Gamma(u_a) \cdot z_a$$

is valid for  $\text{UFP}^{\text{FC}}$  as well, where  $\pi_{s,t}^\mu := \min\{\sum_{a \in P} \mu_a : P \in \mathcal{P}(K)\}$ .  $\square$

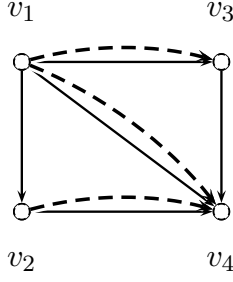
Because  $\text{USPR}^{\text{FC}} \subseteq \text{UFP}^{\text{FC}}$ , the superadditive metric inequalities (8.15) are valid also for  $\text{USPR}^{\text{FC}}$ . As they involve only capacity variables, they are also valid for the polytope  $\text{USPA}^{\text{FC}}$  corresponding to the arc-routing formulation of the fixed change network design problem FC-USPR.

**Corollary 8.15** *For any non-negative metric  $\mu \in \mathbb{R}_+^A$  and any non-decreasing superadditive function  $\Gamma : \mathbb{R} \rightarrow \mathbb{R}$ , the superadditive metric inequality (8.15) is valid for both polytopes  $\text{USPR}^{\text{FC}}$  and  $\text{USPA}^{\text{FC}}$ .*

The following examples shows that the superadditive metric inequalities are indeed stronger than the classical metric inequalities.

**Example 8.2** Let  $D = (V, A)$  be the digraph consisting of the nodes  $V := \{v_1, \dots, v_4\}$  and the arcs  $A := \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_2, v_4), (v_3, v_4)\}$  with capacities  $u_a := 1$  for all  $a \in A$ . Within this digraph, consider the commodities  $K := \{(v_1, v_3), (v_1, v_4), (v_2, v_4)\}$  with demand values are  $d_{s,t} := 0.6$  for all  $(s, t) \in K$ .  $D$  and  $K$  are illustrated in Figure 8.2. Let's consider the fixed change network design problem, where each arc capacity can be installed at most once.

It is obvious that there exists a fractional multicommodity flow routing of the commodities  $K$  within the subgraph  $(V, B)$  with  $B :=$



$$d_{v_1, v_3} = d_{v_1, v_4} = d_{v_2, v_4} = 0.6$$

$$u_a = 1 \text{ for all } a \in A$$

**Figure 8.2** Digraph and Commodities of Example 8.2.2: Solid lines are arcs, dashed lines are commodities

$\{(v_1, v_2), (v_1, v_3), (v_2, v_4), (v_3, v_4)\}$ : One could, for example, route commodities  $(v_1, v_3)$  and  $(v_2, v_4)$  unsplit along the direct arcs  $(v_1, v_3)$  and  $(v_2, v_4)$ , respectively, and route commodity  $(v_1, v_4)$  half along the path  $(v_1, v_2, v_4)$  and half along  $(v_1, v_3, v_4)$ . Hence, all classic metric inequalities (8.12) are satisfied for  $z := \chi(B)$ .

On the other hand, the given capacities do not admit an unsplittable flow routing of the given commodities within the subgraph  $(V, B)$ . In fact, there is a simple violated superadditive metric inequality: Let  $\mu \in \mathbb{R}_+^A$  be the metric given by the cut  $\delta(\{v_1, v_2\})$ , i.e.,  $\mu_{(v_1, v_3)} = \mu_{(v_1, v_4)} = \mu_{(v_2, v_4)} = 1$  and  $\mu_{(v_1, v_2)} = \mu_{(v_3, v_4)} = 0$ . Let  $\Gamma : \mathbb{R} \rightarrow \mathbb{R}$  be defined as  $\Gamma(x) := \lfloor 5/3 \cdot x \rfloor$ . It is easy to verify that  $\Gamma$  is non-decreasing and superadditive. The superadditive metric inequality defined by  $\mu$  and  $\Gamma$  is

$$\Gamma(u_{(v_1, v_3)}) z_{(v_1, v_3)} + \Gamma(u_{(v_1, v_4)}) z_{(v_1, v_4)} + \Gamma(u_{(v_2, v_4)}) z_{(v_2, v_4)} =$$

$$z_{(v_1, v_3)} + z_{(v_1, v_4)} + z_{(v_2, v_4)} \geq \sum_{(s, t) \in K} \Gamma(d_{s, t}) = 3.$$

For  $z := \chi(B)$ , this inequality is clearly violated.

Note that this superadditive metric inequality is stronger than any band, 2-band, or strengthened metric inequality that could be derived from the (fractional) multicommodity flow polytope  $\text{MCF}^{\text{FC}}$ . In fact, this inequality is not valid  $\text{MCF}^{\text{FC}}$ .  $\triangle$

In general, the superadditive metric inequalities (8.15) are not sufficient to characterize all capacity installations that admit an unsplittable flow routing. It is not difficult to construct instances of the fixed charge network design problem with unsplittable flow routing, for which the linear program

$$\min \{ w^T z : (x, z) \in [0, 1]^{\mathcal{P}(K) \cup A} \}$$

$$(x, z) \text{ satisfies (7.3), (7.4), and (8.15) } \},$$

has a smaller solution value than the corresponding integer program.

**Remark 8.16** For any non-decreasing superadditive function  $\Gamma : \mathbb{R} \rightarrow \mathbb{R}$  and for each arc  $a \in A$ , the strengthened capacity constraint (8.16) used to



derive the superadditive metric inequality (8.15) is just some valid inequality for the 0/1-knapsack polytope

$$KN(a) := \text{conv}\{(x, z_a) \in \{0, 1\}^{\mathcal{P}_a} \times \{0, 1\} : (x, z_a) \text{ satisfies (7.4)}\}$$

defined by the capacity constraint (7.4) on arc  $a$ , where  $\mathcal{P}_a = \{P \in \mathcal{P} : a \in P\}$ .

Hence, each superadditive metric inequality (8.15) is equivalent to or dominated by a linear combination of facet-defining inequalities for these knapsack polytopes and the original model (in-)equalities.

This means that the superadditive metric inequalities (8.15) are redundant if all facet-defining inequalities for these knapsack polytopes (or the more restricted precedence constrained knapsack polytopes discussed in Section 8.3) are added to the LP relaxation.

Nevertheless, these inequalities are often useful in practice, because they operate directly and only on the capacity variables  $z_a$ .

The presented approach can be applied analogously to derive valid inequalities for the corresponding polyhedra of the capacitated network design problem with unsplittable flow routing. Again, the resulting inequalities carry over immediately to the unsplittable shortest path routing polyhedra  $\text{USPR}^{\text{Cap}}$  and  $\text{USPA}^{\text{Cap}}$ .

**Theorem 8.17** *For any non-negative metric  $\mu \in \mathbb{R}_+^A$  and any non-decreasing superadditive function  $\Gamma : \mathbb{R} \rightarrow \mathbb{R}$ , the **superadditive metric inequality***

$$\sum_{a \in A} (-\mu_a \Gamma(-u_a)) z_a \geq \sum_{(s,t) \in K} \pi_{s,t}^\mu \Gamma(d_{s,t})$$

*is valid for both polyhedra  $\text{USPR}^{\text{Cap}}$  and  $\text{USPA}^{\text{Cap}}$ , where  $\pi_{s,t}^\mu$  is the length of a shortest  $(s, t)$ -path with respect to  $\mu$ .*

**Proof.** Analogous to the proof of Theorem 8.14. □

For the polyhedra associated with the congestion minimization problems, the strengthening technique of applying a superadditive function to the capacity constraints cannot be used directly, because in the corresponding capacity constraints involve the fractional congestion variable  $L$ . Nevertheless, one might use it indirectly to generate a lower bound on the congestion variable  $L$  by checking for violated strengthened metric inequalities for several fixed threshold congestion values  $\hat{L}$ .

**Theorem 8.18** *Let  $\hat{L} \in \mathbb{R}_+$ . Furthermore, let  $\mu \in \mathbb{R}_+^A$  be a non-negative metric,  $\Gamma : \mathbb{R} \rightarrow \mathbb{R}$  be a non-decreasing superadditive function, and denote by  $\pi_{s,t}^\mu$  the length of a shortest  $(s, t)$ -path with respect to  $\mu$ . If*

$$\sum_{a \in A} \mu_a \Gamma(\hat{L} u_a) < \sum_{(s,t) \in K} \pi_{s,t}^\mu \Gamma(d_{s,t}) ,$$

then the inequality

$$L \geq \hat{L}$$

is valid for  $\text{USPR}^{\text{Con}}$  and  $\text{USPA}^{\text{Con}}$ .

**Proof.** Follows directly from Theorem 8.14.  $\square$

### 8.2.3 Separation of Superadditive Metric Inequalities

The separation problem for superadditive metric inequalities is  $\mathcal{NP}$ -hard in general. Using a construction similar to that in the proof of Theorem 6.1, the PARTITION problem [96, 121] can be reduced to the problem of finding a violated superadditive metric inequality in a ring-network. Even if the metric  $\mu$  is given, the problem of finding a non-decreasing superadditive function  $\Gamma$  that yields the most violated superadditive metric inequality (8.15) remains  $\mathcal{NP}$ -hard.

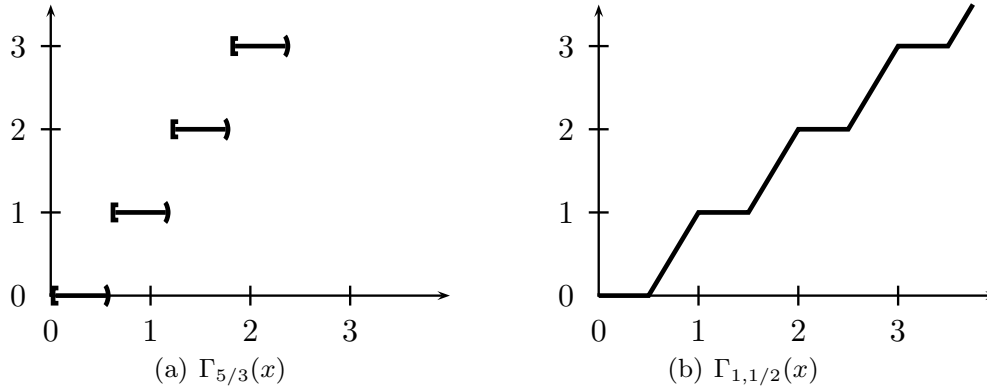
A natural heuristic approach for the separation of superadditive metric inequalities (8.15) is to first generate a metric for which the classical metric inequality (8.12) is as tight as possible (up to scaling of the metric) and then, in a second step, compute a non-decreasing superadditive function  $\Gamma$  that together with  $\mu$  yields a violated inequality (8.15).

The first subproblem, which is nothing else but the separation problem for the classical metric inequalities, can be easily solved in polynomial time.

The second subproblem of finding the best possible general non-decreasing superadditive function  $\Gamma : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  is computationally hard. Therefore, it is useful to retreat to a subclass of these functions such that the best function in this subclass can be computed efficiently.

Belaidouni and Ben-Ameur [20] propose two subclasses of non-decreasing superadditive step functions for this approach. The best function in the first class can be computed in polynomial time using a linear programming approach. For choosing a function from the larger second class, Belaidouni and Ben-Ameur propose a heuristic.

In our implementation, we use a very simple class of non-decreasing superadditive functions. It consists of the step functions  $\Gamma_c(x) = \lfloor cx \rfloor$  for a small number of values  $c \in \mathbb{R}_+$ . Applying such a function to a given inequality corresponds to the standard integer rounding procedure, see Nemhauser and Wolsey [146]. Figure 8.3(a) illustrates the function  $\Gamma_{5/3}(x)$ , which was also used in Example 8.2.2. The strengthened capacity constraints (8.16) obtained by these functions have a very simple interpretation: Regarding  $c$  as a base capacity unit, the values  $\Gamma_c(d_{s,t})$  express how many (integer multiples) of these units are sent with each commodity  $(s, t)$ , and the value  $\Gamma_c(u_a)$  expresses how many (integer multiples) of these units can be fully accommodated within the given arc capacity of  $u_a$ .



**Figure 8.3** Non-decreasing superadditive functions corresponding to integer and strengthened integer rounding.

Another class of simple non-decreasing superadditive functions, which might be useful but is not implemented yet, is the class of functions

$$\Gamma_{c,\alpha}(x) := \begin{cases} \lfloor cx \rfloor, & \text{for } cx - \lfloor cx \rfloor \leq \alpha, \\ \lfloor cx \rfloor + \frac{cx - \lfloor cx \rfloor - \alpha}{1 - \alpha}, & \text{for } cx - \lfloor cx \rfloor > \alpha \end{cases}$$

with  $c \in \mathbb{R}_+$  and  $0 \leq \alpha < 1$ . These functions correspond to the strengthened integer rounding procedure, see Nemhauser and Wolsey [146]. Figure 8.3(b) illustrates the function  $\Gamma_{c,\alpha}(x)$  for  $c = 1$  and  $\alpha = 1/2$ .

### 8.3 Precedence Constrained Knapsack Inequalities

In this section, we consider the subproblem that is defined by a single arc capacity constraint and the routing paths across that arc.

For unsplittable flow problems with fixed or variable arc capacities, each such subproblem is a standard 0/1-knapsack with a fixed or variable knapsack capacities, and valid inequalities for the corresponding knapsack polytopes carry over immediately to valid inequalities for the original unsplittable flow polyhedra.

For unsplittable shortest path routing problems, the interdependencies among the routing paths introduce additional precedence relations among the items in the knapsack problem. For each arc, the subproblem defined by its capacity constraint and the routing paths across this arc can be regarded as a precedence constrained knapsack with fixed or variable capacity. Valid and facet-defining inequalities for the corresponding precedence constrained knapsack polyhedra carry over naturally to valid inequalities for the underlying unsplittable shortest path routing polyhedron. As not only the capacity constraints but also the precedences among the routing paths are respected, the inequalities derived from facets of the precedence constrained knapsack polyhedra are typically much stronger than those derived from facets of the corresponding standard knapsack polyhedra.

In order to illustrate the basic concepts, we focus on the precedence constraint knapsack polytope that arises as relaxation of the path routing formulation of the fixed charge network design problem FC-USPR. For this formulation, the resulting precedence constrained knapsacks have a fixed capacity and a precedence relations are very easy to describe. For the arc routing formulation of FC-USPR, the presented techniques and results apply analogously; only the structure of the underlying precedence relations changes. The formulations of the capacitated network design problem CAP-USPR and of the congestion minimization problem MIN-CON-USPR lead to precedence constrained knapsack problems with general integer or general fractional capacities. Using standard mixed-integer linear programming techniques, the concepts presented throughout this section can be extended to derive valid inequalities for these polytopes as well. Similar extensions have been discussed for the standard knapsack polytope with integer capacities by Brockmüller et al. [43], van Hoesel et al. [188] and for the standard knapsack polytope with a single fractional (capacity) variable by Marchand and Wolsey [138].

### 8.3.1 Basics

Throughout this section, let  $a \in A$  be an arbitrary but fixed arc. For simplicity, we denote by  $\mathcal{P}_a := \{P \in \mathcal{P}(K) : a \in P\}$  the set of routing paths that contain this arc, by  $d_P := d_{s_P, t_P}$  the demand associated with each path  $P \in \mathcal{P}_a$ , and by  $d(S) := \sum_{P \in S} d_P$  the total demand associated with each path set  $S \subseteq \mathcal{P}_a$ .

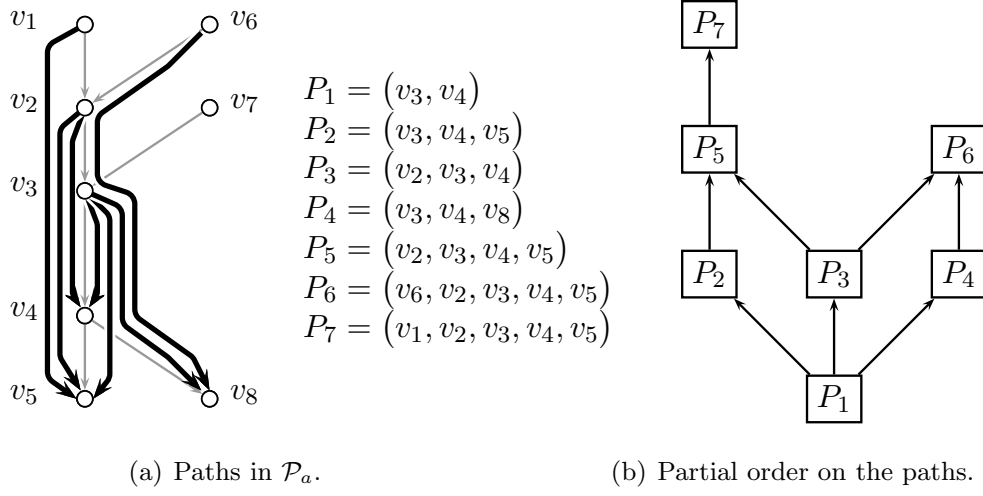
Let  $(x, z) \in \{0, 1\}^{\mathcal{P}(K) \cup A}$  be an integer vector that satisfies the capacity constraints (7.4). If  $z_a = 0$ , then we trivially have  $x_P = 0$  for all  $P \in \mathcal{P}_a$ . Otherwise, if  $z_a = 1$ , the path routing variables  $x_P$  satisfy the capacity constraint

$$\sum_{P \in \mathcal{P}_a} d_P x_P \leq u_a . \quad (8.17)$$

Inequality (8.17) defines a standard 0/1-knapsack. The items of the knapsack are the paths  $P \in \mathcal{P}_a$ , the item sizes are the demand values  $d_P$  associated with these paths  $P \in \mathcal{P}_a$ , and the knapsack capacity is the arc capacity  $u_a$ . It is easy to see that any valid inequality  $\alpha^T x \leq \alpha_0$  for the associated *knapsack polytope*

$$\text{KN}(\mathcal{P}_a, d, u_a) := \text{conv} \{ x \in \{0, 1\}^{\mathcal{P}_a} : x \text{ satisfies (8.17)} \} .$$

carries over to a valid inequality  $\alpha^T x \leq \alpha_0 z_a$  for the polytope  $\text{UFP}^{\text{FC}}$  associated with the fixed charge network design problem with unsplittable flow routing, which was introduced in the previous section. Under some mild conditions, facet-defining inequalities for  $\text{KN}(\mathcal{P}_a, d, u_a)$  yield also facet-defining inequalities for  $\text{UFP}^{\text{FC}}$ .



**Figure 8.4** Precedence relations among the paths in  $\mathcal{P}_a$ .

Since  $\text{USPR}^{\text{FC}} \subseteq \text{UFP}^{\text{FC}}$ , these inequalities are also valid for  $\text{USPR}^{\text{FC}}$ . However, they are typically not very strong, because the interdependencies among the routing paths of an unsplittable shortest path routing are completely disregarded in the standard 0/1-knapsack polytope. Stronger inequalities for  $\text{USPR}^{\text{FC}}$  can be obtained by considering the capacity constraint (8.17) and (some of) the interdependencies among the paths across  $a$  together. The path set  $S(x) := \{P : x_P = 1\}$  defined by an integer vector  $(x, z) \in \text{USPR}^{\text{FC}}$  is an unsplittable shortest routing and thus satisfies the Bellman property, see Definition 5.5 on page 75. This implies that the **precedence constraints**

$$x_Q \geq x_P \quad \text{for all } Q \subset P \quad (8.18)$$

hold for any such vector. Note that these constraints are just linear combinations of the original circuit equalities (7.2), demand equalities (7.3), and the non-negativity constraints  $x_P \geq 0$ . They are clearly valid for  $\text{USPR}^{\text{FC}}$  (and even for its LP relaxation).

The precedence constraints (8.18) define a partial order on the paths in  $\mathcal{P}_a$ . Figure 8.4 illustrates this order for a simple example. Together with the knapsack constraint (8.17), these inequalities define the **precedence constrained knapsack polytope**

$$\text{PCKN}(\mathcal{P}_a, d, u_a) := \text{conv} \{x \in \{0, 1\}^{\mathcal{P}_a} : x \text{ satisfies (8.17) and (8.18)}\}$$

associated with arc  $a$ . For notational simplicity, we denote this polytope by PCKN in the following.

Clearly, any valid inequality for PCKN carries over to a valid inequality for  $\text{USPR}^{\text{FC}}$ .

**Observation 8.19** *If  $\alpha^T x \leq \alpha_0$  is a valid inequality for PCKN, then the inequality  $\alpha^T x \leq \alpha_0 z_a$  is valid for  $\text{USPR}^{\text{FC}}$ .*

Note that in general facet-defining inequalities for PCKN not necessarily yield facet-defining inequalities for USPR. The non-Bellman-type interdependencies among the paths in  $\mathcal{P}_a$  and the generalized upper bound constraints  $\sum_{P \in \mathcal{P}(s,t) \cap \mathcal{P}_a} x_P \leq 1$  implied by the demand constraints (7.3) are still disregarded in PCKN. Nevertheless, facets of PCKN typically yield much stronger valid inequalities for USPR<sup>FC</sup> than the facets of KN.

In the remainder of this section, we review some basic results concerning the facial structure of PCKN, discuss several classes of valid and facet-defining inequalities for PCKN, and show that the precedence constraint knapsack problem – and thus several lifting problems related to these inequalities – can be solved in pseudo-polynomial time for some special case of our application. For simplicity, we formulate all concepts in terms of the paths used in our concrete setting instead of general ‘items’.

### 8.3.2 Properties and Valid Inequalities for PCKN

For any set  $S \subseteq \mathcal{P}_a$ , we denote by

$$l(S) := \{P \in \mathcal{P}_a : P \subseteq Q \in S\}$$

the set of all paths that are subpaths of some path in  $S$ , including the paths in  $S$  themselves. If all paths in  $S$  are chosen as routing paths in an unsplittable shortest path routing, then the Bellman property implies that also all other paths in  $l(S)$  must be chosen. Analogously, we denote by

$$u(S) := \{P \in \mathcal{P}_a : P \supseteq Q \in S\}$$

the set of all paths that are superpaths of some path in  $S$ , including the paths in  $S$  themselves. The sets  $l(S)$  and  $u(S)$  are the lower and the upper ideal of  $S$  in the partial order defined by the Bellman property. For notational simplicity, we write just  $l(P)$  for  $l(\{P\})$  and  $u(P)$  for  $u(\{P\})$ . According to the standard terminology for partial ordered set, two paths  $P$  and  $Q$  are said to be *incomparable* if neither  $P \subset Q$  nor  $Q \subset P$ . A path set  $S \subseteq \mathcal{P}_a$  is called *incomparable* if all its paths are pairwise incomparable.

Without loss of generality, we assume in the following that

$$d(l(P)) \leq u_a \quad \text{for all } P \in \mathcal{P}_a, \tag{8.19}$$

that is, every single path and all its subpaths fit into the knapsack capacity. If (8.19) does not hold for some path  $P$ , then  $x_P = 0$  for all  $x \in \text{PCKN}$ . In this case, we can remove the path  $P$  from the precedence constraint knapsack and consider only the face of PCKN that is induced by  $x_P = 0$ , which is equivalent to the smaller precedence constrained knapsack polytope  $\text{PCKN}(\mathcal{P}_a \setminus \{P\}, d, u_a)$ . Facets of this smaller polytope can be lifted trivially to facets of the original one.

If condition (8.19) holds, one easily observes the following.

**Theorem 8.20 (Boyd [42])**

- (i) PCKN is full-dimensional, i.e.,  $\dim(\text{PCKN}) = |\mathcal{P}_a|$ .
- (ii)  $x_P \geq 0$  is a facet of PCKN if and only if  $u(P) = \{P\}$ .
- (iii)  $x_P \leq 1$  is a facet of PCKN if and only if
  - $l(P) = \{P\}$  and
  - $d(l(P') \cup \{P\}) \leq u_a$  for all  $P' \in \mathcal{P}_a$ .
- (iv)  $x_{P'} \geq x_P$  is a facet of PCKN if and only if
  - $P' \subset P$ ,  $P' \neq P$ , and
  - $d(l(Q) \cup l(P)) \leq u_a$  for all  $Q$  with  $P' \in l(Q)$ .

Most of the valid and facet-defining inequalities for PCKN studied in the literature are strengthened versions of well-known inequalities for the standard 0/1-knapsack polytope.

The first class of valid inequalities for PCKN that we discuss here is a generalization of the minimal cover inequalities for the standard 0/1-knapsack problem, which have been introduced independently by Balas [13], Padberg [152], Hammer et al. [111], and Wolsey [192].

**Definition 8.21 (Boyd [42])** A set  $C \subseteq \mathcal{P}_a$  is called an *induced cover* if  $d(l(C)) > u_a$ .

An induced cover is a set of paths that together with their subpaths exceed the capacity of the knapsack. For any such induced cover  $C \subseteq \mathcal{P}_a$ , the **induced cover inequality**

$$\sum_{P \in C} x_P \leq |C| - 1 \tag{8.20}$$

is valid for PCKN. In order to be facet-defining for PCKN, an induced cover inequality must not be dominated by another induced cover inequality. This leads straightforward to the notion of minimal induced covers.

**Definition 8.22 (Boyd [42])** A set  $C \subseteq \mathcal{P}_a$  is said to be a *minimal induced cover* if

- (i)  $C$  is incomparable,
- (ii)  $C$  is an induced cover, i.e.,  $d(l(C)) > u_a$ , and
- (iii)  $d(l(C) \setminus \{P\}) \leq u_a$  for each  $P \in C$ .

In our setting, a minimal induced cover is a set of incomparable paths that together with their induced subpaths exceed the given capacity, whereas all of the induced subpaths and all but one of the original paths fit into the capacity.

**Theorem 8.23 (Boyd [42], Park and Park [157])** *Given a minimal induced cover  $C \subseteq \mathcal{P}_a$ , the induced cover inequality (8.20) defines a facet of PCKN if and only if  $l(P) \cap l(P') = \emptyset$  for all  $P, P' \in C$ .*

For an arbitrary minimal induced cover  $C$ , the induced cover inequality (8.20) is facet-defining for

$$\text{conv} \left\{ x \in \text{PCKN} : \begin{array}{ll} x_P = 0 & \text{for } P \in \mathcal{P}_a \setminus l(C), \text{ and} \\ x_P = 1 & \text{for } P \in l(C) \setminus u(C) \end{array} \right\},$$

which is a face of PCKN. Via sequential lifting, any minimal induced cover inequality can be turned into facets for the entire polytope.

It is easy to see that, for any minimal induced cover  $C$ , inequality (8.20) dominates the standard 0/1-knapsack cover inequality

$$\sum_{P \in l(C)} x_P \leq |l(C)| - 1,$$

which could be derived from the corresponding knapsack polytope KN when disregarding the precedence constraints among the paths in  $l(C)$ .

Another class of valid inequalities is derived from a simple generalization of minimal induced covers.

**Definition 8.24 (Boyd [42])** *Let  $k \in \mathbb{Z}_+$  with  $k \geq 2$ . A set  $C \subseteq \mathcal{P}_a$  is said to be an **induced  $k$ -cover** if*

- (i)  $C$  is incomparable, and
- (ii) each subset  $S \subseteq C$  with  $|S| = k$  is a minimal induced cover, i.e.,  $d(l(S)) > u_a$  and  $d(l(S) \setminus \{P\}) \leq u_a$  for each  $P \in S$ .

The induced  $k$ -covers  $C$  with  $k = |C|$  are precisely the minimal induced covers. It is easy to verify that, for each induced  $k$ -cover  $C \subseteq \mathcal{P}_a$ , the **induced  $k$ -cover inequality**

$$\sum_{P \in C} x_P \leq k - 1 \tag{8.21}$$

is valid for PCKN. The following theorem characterizes the conditions under which the induced  $k$ -cover inequality (8.21) defines a facet of PCKN.



**Theorem 8.25 (Boyd [42])** *Given an induced  $k$ -cover  $C \subseteq \mathcal{P}_a$ , the induced  $k$ -cover inequality (8.21) defines a facet of PCKN if and only if*

$$\bigcap_{S \subseteq C: |S|=k-1} l(S) = \emptyset .$$

A sequential lifting procedure that turns the  $k$ -cover inequality of an arbitrary  $k$ -cover  $C$  into a facet-defining inequality was proposed by van de Leensel et al. [187].

Also  $(1, k)$ -configuration inequalities, which have been introduced by Padberg [153] for the standard 0/1-knapsack polytope, can be strengthened by exploiting the given precedence constraints.

**Definition 8.26 (Boyd [42])** *Let  $C \subseteq \mathcal{P}_a$ ,  $P \in \mathcal{P}_a \setminus C$ , and  $k \in \mathbb{Z}_+$  with  $2 \leq k \leq |C|$ . The pair  $(C, P)$  is called an **induced  $(1, k)$ -configuration** if*

- (i)  $C \cup \{P\}$  is incomparable,
- (ii)  $C \cup \{P\}$  is an induced cover, i.e.,  $d(l(C \cup \{P\})) > u_a$ ,
- (iii)  $d(l(C \cup \{P\}) \setminus \{P\}) \leq u_a$ , and
- (iv) for each  $S \subseteq C$  with  $|S| = k$ , the set  $S \cup \{P\}$  is a minimal induced cover.

Again, it is not difficult to verify that, for every induced  $(1, k)$ -configuration  $(C, P)$  and every subset  $Z \subseteq C$  with  $r := |Z| \geq k$ , the **induced  $(1, k)$ -configuration inequality**

$$(r - k + 1) x_P + \sum_{Q \in Z} x_Q \leq |r| \tag{8.22}$$

is valid for PCKN. Boyd [42] has shown that this inequality is facet-defining under the following conditions.

**Theorem 8.27 (Boyd [42])** *Let  $(C, P)$  be an induced  $(1, k)$ -configuration and  $Z \subseteq C$  with  $r := |Z| \geq k$ . Then the induced  $(1, k)$ -configuration inequality (8.22) defines a facet of PCKN if and only if*

- (i)  $l(P) \cap l(Z) = \emptyset$ , and
- (ii)  $\bigcap_{S \subseteq Z: |S|=k-1} l(S) = \emptyset .$

For those induced  $(1, k)$ -configurations  $(C, P)$  that do not satisfy these conditions, the corresponding induced  $(1, k)$ -configuration inequality (8.22) defines a facet of

$$\text{conv}\{x \in \text{PCKN} : x_Q = 0 \text{ for } Q \in (\mathcal{P}_a \setminus l(C \cup \{P\})) \cup (C \setminus Z), \text{ and} \\ x_Q = 1 \text{ for } Q \in l(C \cup \{P\}) \setminus u(C \cup \{P\})\} ,$$

which is a face of PCKN. Again, these inequalities can be turned into facets of the entire precedence constrained knapsack polytope PCKN via sequential lifting.

The separation problem for all three presented classes of inequalities is  $\mathcal{NP}$ -hard in general. In our implementation, we use a set of simple greedy heuristics to find violated minimal induced cover inequalities (8.20).

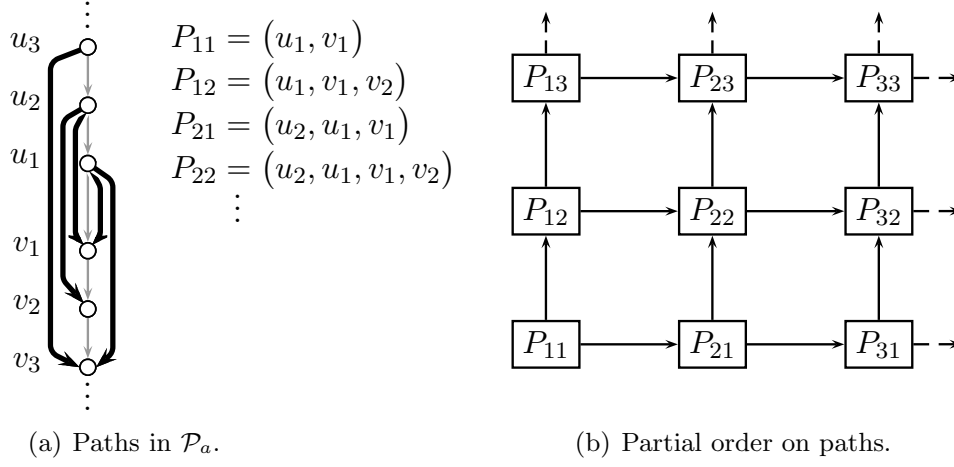
### 8.3.3 Lifting

In many cases, the inequalities discussed above define only a facet of a face of PCKN. When using a standard sequential lifting procedure to turn them into inequalities that are facet-defining for the entire polytope PCKN, the problem of determining the lifting coefficient for a single variable is a precedence constrained knapsack problem with the same precedence relations as the underlying original problem. Garey and Johnson [96] have shown that this problem is strongly  $\mathcal{NP}$ -complete in general.

Van de Leensel et al. [187] introduced a special class of lifting orders, so-called *predecessor first remaining variables second (PFRS)* orders. When using these orders to sequentially lift a minimal induced cover or an induced  $(1, k)$  configuration inequality  $\alpha^T x \leq \alpha_0$ , the lifting coefficients of the predecessor variables  $x_P$ ,  $P \in l(\text{sup}(\alpha))$ , can be determined in pseudo-polynomial time. The computation of the lifting coefficients for the remaining variables as well as the computation of the lifting coefficients for induced  $k$ -cover inequalities remains strongly  $\mathcal{NP}$ -complete in general also for these lifting orders.

For certain special cases of precedence relations, however, the precedence constrained knapsack problem – and thus also the lifting problem for all variables – is solvable in pseudo-polynomial time. Johnson and Niemi [120] propose a dynamic programming algorithm for the case where the precedence relations form a tree. This algorithm generalizes straightforward to the case where precedence relations form a graph with bounded tree-width.

The precedence graphs arising from the Bellman property in our application unfortunately have unbounded tree-width. Even if we consider only a single path  $P$  and all its subpaths across arc  $a$ , the precedence relations form a grid, whose tree-width depends of the length of  $P$ , see Figure 8.5. Nevertheless, also this special case of the precedence constrained knapsack



**Figure 8.5** Precedence relations among the paths in  $\mathcal{P}_a$  in the case where  $\max\{\delta^+(v), \delta^-(v) : v \in V\} = 1$ .

problem can be solved in pseudo-polynomial time. Let  $\delta_{\max} := \max\{\delta(v) : v \in V\}$  denote the maximum degree in the digraph  $D = (V, A)$  of the underlying unsplittable shortest path routing problem. Then we have the following result.

**Theorem 8.28** *For any constant  $k \in \mathbb{Z}_+$ , the precedence constrained knapsack problem  $\{\max \alpha^T x : x \in \text{PCKN}(\mathcal{P}_a, d, u_a)\}$  can be solved in pseudo-polynomial time for all instances of FC-USPR with  $\delta_{\max} \leq k$ , where  $a \in D$  and  $\alpha \in \mathbb{Z}_+^{\mathcal{P}_a}$  is an arbitrary non-negative objective function.*

In the following, we present a dynamic programming algorithm to solve this precedence constrained knapsack problem in the special case where both the maximum in-degree and the maximum out-degree are bounded by 1. This basic algorithm can be extended to the case where the degree in the underlying digraph is bounded by an arbitrary constant by incorporating the subtree decomposition ideas that are used in the dynamic programming algorithms of Johnson and Niemi [120] and Flippo et al. [90] for the tree knapsack problem. Because this generalization is extremely technical and some serious notational overhead would be required to formulate it mathematically correct, it is omitted here.

**Proof** [Theorem 8.28 for  $\max\{\delta^+(\mathbf{v}) : \mathbf{v} \in \mathbf{V}\} \leq 1$  and  $\max\{\delta^-(\mathbf{v}) : \mathbf{v} \in \mathbf{V}\} \leq 1$ ]. If both the maximum in-degree and the maximum out-degree of the underlying digraph are bounded by 1, then all paths in  $\mathcal{P}_a$  are subpaths of either a common directed superpath or a common directed cycle. Without loss of generality, we may assume that all paths in  $\mathcal{P}_a$  are subpaths of a common superpath  $P = (u_m, \dots, u_1, v_1, \dots, v_m)$  with  $a = (u_1, v_1)$ , as illustrated in Figure 8.5.

Furthermore, we may assume that  $\mathcal{P}_a$  consists of all subpaths  $P' \subset P$  with  $a \in P'$ . If this is not the case, we simply let  $P$  be the shortest path that contains all paths given in  $\mathcal{P}_a$  and then add all missing subpaths  $P'$  of  $P$

with cost  $\alpha_{P'} = 0$  to  $\mathcal{P}_a$ . As  $P$  contains at most  $2 \cdot \max\{|P'| : P' \in \mathcal{P}_a\}$  many arcs, the encoding size of the extended problem instance remains polynomial in the original instance's encoding size.

For each pair of nodes  $u_i$  and  $v_j$ , we denote by  $P_{ij}$  the  $(u_i, v_j)$ -subpath of  $P$ . With the above assumption, we have  $\mathcal{P}_a = \{P_{ij} : 1 \leq i \leq m, 1 \leq j \leq n\}$ . As shown in Figure 8.5, the (transitively reduced) precedence graph among these paths is a directed grid of size  $m \times n$ .

For each pair  $(k, l)$  with  $1 \leq k \leq m$  and  $1 \leq l \leq n$  and for each  $U$  with  $1 \leq U \leq u_a$ , we let

$$\begin{aligned} F(k, l, U) := \{ \max \alpha^T x : x \in \{0, 1\}^{\mathcal{P}_A} , \\ \sum_{P \in \mathcal{P}_a} d_P \leq U , \\ x_{P_{k,l}} = 1 , \\ x_{P_{k,j}} = 0 \quad \text{for all } j > l, \\ x_{P_{i,j}} = 0 \quad \text{for all } i < k \text{ and all } j, \\ x_{P_{i,j}} \geq x_{P_{i+1,j}} \quad \text{for all } i \geq k \text{ and all } j, \\ x_{P_{i,j}} \geq x_{P_{i,j+1}} \quad \text{for all } i \geq k \text{ and all } j \} . \end{aligned}$$

According to this definition,  $F(k, l, U)$  is the optimal solution value of the precedence constrained knapsack (sub-)problem that is given by the subset of paths  $P_{ij}$  with  $i \geq k$ , and assuming in addition that path  $P_{kl}$  is chosen, that path  $P_{kl+1}$  is not chosen, and that the knapsack capacity for the remaining (non-fixed) paths is  $U$ .

One easily observes that

$$\{ \max \alpha^T x : x \in \text{PCKN} \} = \max \{ F(m, l, u_a) : 1 \leq l \leq n \} . \quad (8.23)$$

Furthermore, it is not difficult to verify that  $F$  obeys the following recursion. For  $k = m$  and for all  $l = 1, \dots, n$  and all  $U = 0, \dots, u_a$ :

$$F(m, l, U) = \begin{cases} \sum_{i \leq l} \alpha_{P_{mi}} & \text{if } \sum_{i \leq l} d_{P_{mi}} \leq U \\ 0 & \text{otherwise.} \end{cases}$$

For all  $k = 1, \dots, m-1$  and for all  $l = 1, \dots, n$  and all  $U = 0, \dots, u_a$ :

$$\begin{aligned} F(k, l, U) = \max \Big\{ F(k+1, j, U') + \sum_{i \leq l} \alpha_{P_{ki}} : j, U' \text{ with} \\ 1 \leq j \leq l \text{ and } U' := U - \sum_{i \leq l} d_{P_{ki}} \geq 0 \Big\} . \end{aligned}$$

Using a standard dynamic programming algorithm based on this recursion, all function values of  $F$  in the domain  $\{1, \dots, m\} \times \{1, \dots, n\} \times \{0, \dots, u_a\}$  can be computed in pseudo-polynomial time. With (8.23), this implies that the given precedence constrained knapsack problem  $\{ \max \alpha^T x : x \in \text{PCKN} \}$  can be solved in pseudo-polynomial time.  $\square$

---

**Remark 8.29** With a slight modification, the dynamic programming algorithm presented above can also be used to solve the precedence constrained knapsack problem for precedence graphs that are (linear) interval intersection graphs or circular interval intersection graphs. So, also these special cases of the precedence constrained knapsack problem are solvable in pseudo-polynomial time.



## Chapter 9

# Implementation and Computational Results

In this final chapter we describe the algorithm that is used to solve the network design and routing optimization problems arising in the planning and operation of the IP backbone networks of the German national research network. The algorithm combines and extends the concepts and techniques exposed throughout this thesis. It is based on the general solution approach illustrated at the beginning of Chapter 7: In the first phase, a (near-) optimal set of end-to-end routing paths, together with a network configuration is computed using integer linear programming techniques. Afterwards, a valid routing metric that induces exactly these optimal paths is computed.

The integer linear programming models used in the first phase of our algorithm are based on those presented in Chapter 7, but they incorporate more of the technical details and operational side constraints that arise in real-world IP network planning. After the end-to-end paths are found, the compatible routing metric is computed with the linear programming and rounding algorithms described in Chapter 4 of this thesis.

In Section 9.1 we describe the more realistic problem setting implemented in our algorithm and present the corresponding integer linear models. This specific problem has been set up in cooperation with the DFN-Verein for the purpose of planning of the German national research networks G-WiN and X-WiN. Although tailored to fit needs of the DFN-Verein, the models are applicable for a wide range of IP networks and IP network planning problems. They contain a very detailed yet very general description of the link and node hardware used in practice, and they also incorporate several types of typical routing restrictions that arise in real-world routing problems.

Section 9.2 is dedicated to the computational aspects of our implementation. Here we discuss the general solution approach and describe which methods and strategies are used for separation, column generation, branch node and branch variable selection. We also sketch the heuristics that are used to compute feasible solutions in the branch and bound algorithm. Some

of the details that proved to be crucial for the performance of the overall algorithm are discussed in this section too.

In Section 9.3 we report and discuss the computational results obtained with this algorithm for the DFN-Verein planning problems and for some test instances from the SNDlib [149]. We conclude this chapter by discussing some of the strengths and weaknesses of the implemented algorithm and by pointing out some directions for potential future improvements.

## 9.1 Modeling the Real Problem

In the sequel we present the mathematical models that were developed for solving the network design and traffic engineering problems for the Germany's national research and education network. We focus on those technical and operational variants that were used in the planning of the backbone network G-WiN, but the models are not specific for the G-WiN. They were designed to capture many aspects of practical interest as general as possible and are they applicable for many other networks as well.

### 9.1.1 Underlying Networks

The point-to-point link technologies used in the G-WiN provide the same routing capacity for both directions of a link. Multi-point links connecting more than two routers via a single shared-capacity line are not used in the G-WiN. For dimensioning purposes, the network therefore is considered as an undirected link graph, whose nodes correspond to the locations where a router is or can be installed and whose edges correspond to the potential links between these routers.

**Notation 9.1** *The topology of the network is given as an undirected **link graph**  $G = (V, E)$ .*

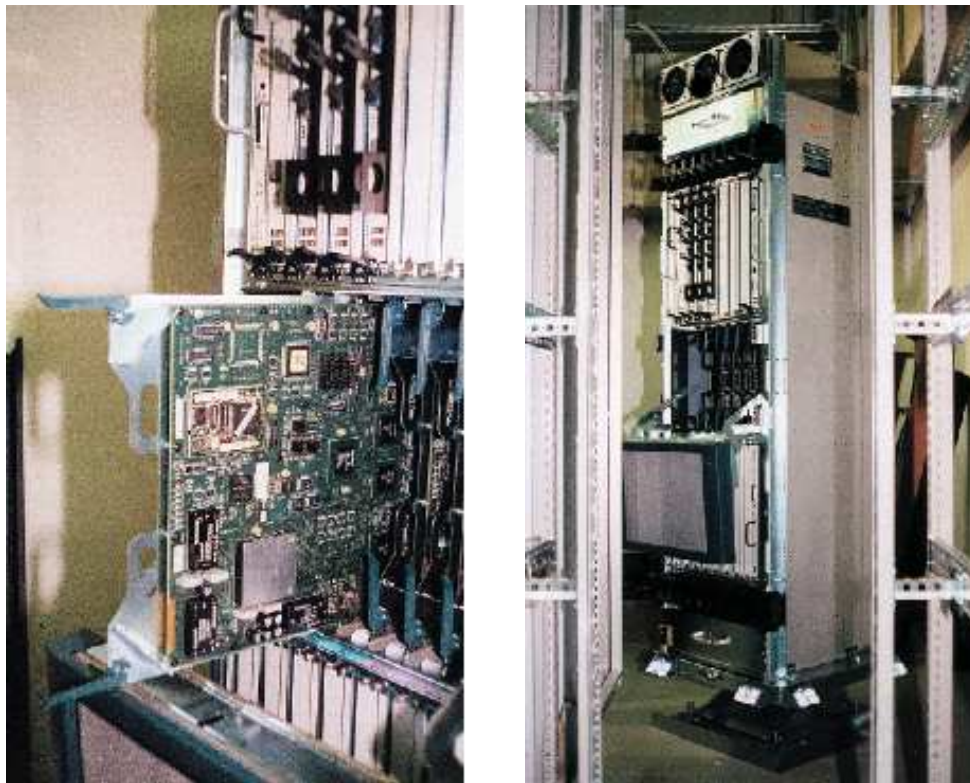
The undirected link graph  $G$  defines the directed supply graph, in which the directed traffic demands are routed.

**Notation 9.2** *With the given link graph  $G = (V, E)$  we associate the directed **supply graph**  $D = (V, A)$ , where*

$$A := \{ (u, v), (v, u) : uv \in E \} .$$

In short-term traffic engineering problems, where the both topology and the hardware configuration of the real network must remain unchanged, we may of course assume that the supply graph and its arc capacities are given directly. In medium- and long-term network design and network reconfiguration problems, we need consider the possible hardware installation in the undirected link graph.



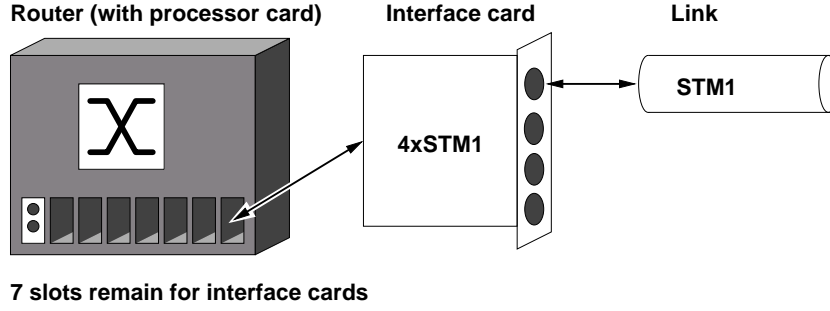


**Figure 9.1** Switching fabric and fully equipped Cisco 12016 Gigabit-router in the G-WiN network [72].

### 9.1.2 Hardware and Network Configuration

For each node and each link in the network, appropriate node hardware or link technology must be installed. In practice, this is not just one single device or capacity type for each node or link. Many different hardware devices and software components are necessary to realize the required functionalities.

Figure 9.1 shows a typical IP router used in the G-WiN and in other networks. The entire router system consists of a core unit and a number of different interface cards. The core unit contains a routing processor card, redundant clock and scheduler cards, a number of switching fabric cards, and redundant power supply for the entire system. The routing processor plus the number and the type of switching fabrics determine the maximum total throughput of the router. In principle, all these cards are separate devices that may be purchased, installed, or replaced independent of each other. Yet, a router system typically can be equipped with only one type of processor, clock, and switching fabric card. Since these devices are necessary for the router to be operational, they can be regarded as an integral part of the core unit here. Figure 9.2 illustrates the simplified architecture of a router, where only the (pre-equipped) core router and the interface cards are considered. There exist various router systems, and interface cards for one



**Figure 9.2** Simplified architecture of an IP router.

system typically cannot be used together with the core unit of another system. In general, the chosen core unit type determines which and how many interface cards, switching fabrics, and routing processors can be installed. Most router vendors offer interface cards for various link technologies, such as SONET, ETHERNET, ATM, or WDM, and for various link capacities. Thus, many different link technologies and capacities can be used within one network.

In the following we present a very generic model for the network configuration and the hardware installation, which was introduced by Kröller [127]. Besides the technical aspects of the hardware installation such as hardware compatibility, many non-technical issues arising in real-world planning problems can be expressed within this model.

The model uses the concept of abstract components and resources. Components are entities that can be installed somewhere in the network in an integer amount. A component may be a real item, such as router or an interface card, as well as an artificial item, like a rebate available when buying bundles of hardware.

**Notation 9.3** Let  $G = (V, E)$  be the link graph. Components can be installed at nodes  $v \in V$ , at edges  $e \in E$ , and at the graph  $G$  itself. The set of these **graph elements** is

$$\mathcal{G} = \mathcal{G}(G) := \{G\} \cup V \cup E.$$

**Notation 9.4** The set of all **components** is denoted by  $C$ . The graph elements themselves are also components, i.e.,  $\mathcal{G} \subseteq C$ .

A component  $c \in C$  has to be installed at least  $\underline{z}_{c,g}$  and at most  $\bar{z}_{c,g}$  times at the graph element  $g \in \mathcal{G}$ , where  $\underline{z}_{c,g}, \bar{z}_{c,g} \in \mathbb{Z}_+$  with  $\underline{z}_{c,g} \leq \bar{z}_{c,g}$ . Each graph element  $g \in \mathcal{G}$  is a special component that is installed exactly once at itself and only there, i.e.,

$$\begin{aligned} \underline{z}_{g,g} &= \bar{z}_{g,g} = 1 && \text{for all } g \in \mathcal{G} \text{ and} \\ \underline{z}_{g,g'} &= \bar{z}_{g,g'} = 0 && \text{for all } g, g' \in \mathcal{G} \text{ with } g \neq g'. \end{aligned}$$

**Notation 9.5** A component  $c \in C$  is **installable** at graph element  $g \in \mathcal{G}$ , if  $\bar{z}_{c,g} > 0$ . A vector  $z = (z_{c,g}) \in \mathbb{Z}_+^{C \times \mathcal{G}}$  is called a **component installation** if

$$\underline{z} \leq z \leq \bar{z} , \quad (9.1)$$

where  $\underline{z} = (\underline{z}_{c,g}) \in \mathbb{Z}_+^{C \times \mathcal{G}}$  and  $\bar{z} = (\bar{z}_{c,g}) \in \mathbb{Z}_+^{C \times \mathcal{G}}$  are the lower and upper installation bounds, respectively.

The side constraints of the component installation are expressed via resources, which are provided or consumed by the installed components. A resource may stand for a technical restriction, such as the type and the number of interface cards that can be plugged into a core router, or for some administrative condition, such as a maximum number of allowed device exchanges in a network reconfiguration.

**Notation 9.6** The set of all **resources** is denoted by  $R$ .

If a component is installed at a graph element, it provides or consumes some of the resources.

**Notation 9.7** For each graph element  $g \in \mathcal{G}$ , each component  $c \in C$  (that is installable at  $g$ ), and each resource  $r \in R$ , the **resource value**  $\varrho_{c,g}^r \in \mathbb{Z}$  denotes the contribution to the resource  $r$  of a single installation of component  $c$  at  $g$ .

If  $\varrho_{c,g}^r > 0$ , then installing one component  $c$  at  $g$  provides  $\varrho_{c,g}^r$  units of  $r$ . If  $\varrho_{c,g}^r < 0$ , then  $|\varrho_{c,g}^r|$  units of  $r$  are consumed by installing  $c$  at  $g$ .

In principle, a resource must not be exhausted, i.e., the consumption of a resource must not exceed the provision. However, there are several different ways how components installed at different graph elements share resources. For example, a total cost budget is a global resource that is consumed by all components installed in the network. The maximum number of interface cards that can be plugged into a core router is a restriction that is only locally valid at each node. Clearly, the free slots in the router's core unit at a node  $v \in V$  are not consumed by the interface cards that are installed at some other node  $w \in V$ ,  $w \neq v$ . To model these differences, we distinguish three types of resources.

**Notation 9.8** The resource set is partitioned into

$$R^L \dot{\cup} R^V \dot{\cup} R^G = R .$$

Each resource  $r \in R$  is either a **local resource**  $r \in R^L$ , a **node resource**  $r \in R^V$ , or a **global resource**  $r \in R^G$ .

For each local resource  $r \in R^L$  and for each graph element  $g \in \mathcal{G}$ , the amount of resource  $r$  consumed by the components installed at  $g$  must not exceed the amount provided by the components at  $g$ . Similar, node resources balance the resource values of the components installed at a node and its adjacent edges, while global resources balance the resource values of all installed components. These constraints can be formulated with the following linear inequalities:

$$\sum_{c \in C} \varrho_{c,g}^r z_{c,g} \geq 0 \quad \text{for all } g \in \mathcal{G}, r \in R^L, \quad (9.2)$$

$$\sum_{g \in \{v\} \cup \delta(v)} \sum_{c \in C} \varrho_{c,g}^r z_{c,g} \geq 0 \quad \text{for all } v \in V, r \in R^V, \quad (9.3)$$

$$\sum_{g \in \mathcal{G}} \sum_{c \in C} \varrho_{c,g}^r z_{c,g} \geq 0 \quad \text{for all } r \in R^G. \quad (9.4)$$

The three types of local, node, and global resources suffice to describe all relevant hardware compatibility constraints and network configuration restrictions arising in the planning of the G-WiN.

**Notation 9.9** A component installation  $z \in \mathbb{Z}_+^{C \times \mathcal{G}}$  is called a *feasible dimensioning*, if it satisfies the bounds (9.1) and all resource inequalities (9.2), (9.3), and (9.4). The set of all feasible dimensionings is

$$\mathcal{Z} := \left\{ z \in \mathbb{Z}_+^{C \times \mathcal{G}} : z \text{ satisfies (9.1), (9.2), (9.3), and (9.4)} \right\}.$$

The components that are installed at the edges of the link graph (may) provide bidirectional routing capacities. This is modeled with a special local resource  $\text{cap} \in R^L$ . We assume that the total routing capacity available on an edge  $e \in E$  is precisely the sum of the capacity contributions  $\varrho_{c,e}^{\text{cap}}$  of all components  $c$  installed at  $e$ . This assumption is realistic, if either at most one (real) component per edge provides capacity in any feasible dimensioning, or if the flow across the link may use the capacities of all components that are installed in parallel. If such flow-sharing technology is not available, then parallel link technologies appear as different links to the IP router and must be modeled by parallel edges in the link graph.





**Notation 9.10** Let  $z \in \mathbb{Z}_+^{C \times \mathcal{G}}$  be a feasible dimensioning. The *(routing) capacities*  $u = (u_a) \in \mathbb{R}_+^A$  provided by  $z$  in the digraph  $D$  are

$$u_{(v,w)} = u_{(w,v)} := \sum_{c \in C} \varrho_{c,vw}^{\text{cap}} z_{c,vw} \quad \text{for all } vw \in E,$$

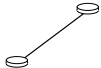


where  $\text{cap} \in R^L$  denotes the special local resource routing capacity.

The cost of a network dimensioning depends on the installed components. It is expressed using another special resource.

**Components installable at node  $v$** 

<b>v</b> Resource values: <b>space</b> 1 	<b>cr-12008</b> Resource values: <b>space</b> -1 <b>slot</b> 6 <b>swcap</b> 10,000 
<b>4xstm1-card</b> Resource values: <b>slot</b> -1 <b>stm1</b> 4 	<b>4xstm4-card</b> Resource values: <b>slot</b> -1 <b>stm4</b> 4 

**Components installable at edge  $e$** 

<b>e</b> Resource values: <b>fiber</b> 1 	<b>stm1-link</b> Resource values: <b>stm1</b> -1 <b>swcap</b> -155 <b>fiber</b> -1 <b>cap</b> 155 	<b>stm4-link</b> Resource values: <b>stm4</b> -1 <b>swcap</b> -622 <b>fiber</b> -1 <b>cap</b> 622 
---	---	---

**Figure 9.3** Typical components and resources considered in an IP network.

**Notation 9.11** The cost of a network dimensioning  $z \in \mathbb{Z}_+^{C \times \mathcal{G}}$  is

$$\sum_{g \in \mathcal{G}} \sum_{c \in C} \varrho_{c,g}^{\text{cost}} z_{c,g} ,$$

where  $\text{cost} \in R^G$  denotes the special global resource cost.

**Example 9.1** Figure 9.3 illustrates some typically components considered in an IP-over-SDH network like G-WiN. The graph node component  $v$  provides the room necessary for the installation of other components at  $v$ . With the local resource **space** it expresses that at most one router may be installed. In this example, we consider only one router system, a Cisco 12008. The core unit of this router system is modeled by the component **cr-12008**, which consumes 1 unit of the resource **space**. In practice, one

may have the choice between several router systems. A Cisco 12008 core unit has a switching capacity of 10 Gbit/s and 8 card slots. One of these slots is occupied by the routing processor card and one is occupied by a scheduler card, 6 slots remain free. This is described with the node resource **swcap** and the local resource **slot**, respectively. Each of the free slots can be equipped with an interface card (or another switching fabric to increase the maximum throughput of the router, but this is ignored in our example). The interface cards provide ports to attach links. In this example, we consider two types of interface cards. One type provides four ports for STM-1 links and is modeled by the component **4xstm1-card**. The other card type provides four STM-4 ports and is modeled as component **4xstm4-card**. The node resources **stm1** and **stm4** express the respective port types.

At the edges, three different components are considered. At each edge  $e$ , component **e** specifies how many and which link capacity types may be installed at  $e$ . We consider two possible capacities STM-1 and STM-4, which are modeled by the components **stm1-link** and **stm4-link**. Since the component **e** provides one unit of the local resource **fiber** and each component **stm1-link** and **stm4-link** consumes one unit, at most one of the two capacity types may be installed at edge  $e$ . Combining these capacities or installing multiples of each is not allowed. If **stm1-link** is installed at edge  $e = uv$ , it consumes one STM-1 port and 155 Mbit/s of the router's switching capacity at each end-node  $u$  and  $v$ . In turn, it provides a bidirectional routing capacity of 155 Mbit/s for the IP traffic across edge  $e$ . Analogously, an **stm4-link** component consumes an STM-4 port and 622 Mbit/s of the router's switching capacity at each end-node and provides a routing capacity of 622 Mbit/s.  $\triangle$

The component-resource model introduced above is very flexible and allows to model the hardware installation part of network planning problems at a very detailed level. The simplified link capacity models used in the basic unsplittable shortest path routing problems MIN-CON-USPR, FC-USPR, and CAP-USPR can be easily expressed in terms of components and resources (if we assume that the same capacity must be installed for both directions of each link).

Besides the pure hardware installation and compatibility, the component-resource model can be used to capture many other aspects that are relevant in practice.

In the planning of the G-WiN, for example, we used artificial components and resources to model pools of freely available hardware and to express the demand of spare ports at the nodes, which were needed to connect the core network to common Internet exchange points and to local networks. In the network expansion and reconfiguration planning, we also used special components and resources to bound the number (or the associated cost) of modifications that can be made to the existing network. Kröller [127]

discusses some of these modeling possibilities in detail.

### 9.1.3 Traffic Demands and Routing

The routing of the data traffic in the G-WiN is controlled by the OSPF protocol. For operational reasons, the DFN-Verein wants all packets with the same origin and destination to follow the same path through the network. Thus, the routing can be modeled as an unsplittable shortest path routing.

**Notation 9.12** *The traffic demands are given as a set of directed **commodities**  $K \subseteq V \times V$ . Each commodity  $(s, t) \in K$  has an associated **demand values**  $d_{s,t} \in \mathbb{Z}_+$ .*

In practice, the routing often has to fulfill several side constraints, which have been disregarded in the pure unsplittable shortest path routing version studied in the preceding chapters. In the G-WiN, we have to consider three kinds of routing restrictions: hop limits, node taboos, and the symmetry requirement. Hop limits bound the number of routers in the admissible routing paths and are introduced to ensure low and even packet delays. Node taboos force some or all routing paths to avoid certain nodes. They are introduced to ensure that certain special nodes have no transit traffic. In some of the G-WiN planning instances, the considered link graph contains ‘artificial’ nodes that correspond to external networks reachable via different gateways or to important nodes at the next lower network level. In both cases, traffic between the normal nodes shall not transit these nodes. Finally, the routing symmetry requirement means that the same routing length must be assigned to the two anti-parallel arcs defined by an undirected edge of the link graph. Consequently, the same path is used for both directions of a communication. This requirement is often introduced to ensure the traceability of data packet flows and improve.

The hop limits and the node taboos are specified via the following parameters.

**Notation 9.13** *For each commodity  $(s, t) \in K$ , we are given a **hop limit**  $\ell_{s,t} \in \mathbb{Z}_+$ , which specifies the maximum number of arcs allowed in the routing path for commodity  $(s, t)$ .*

**Notation 9.14** *For each commodity  $(s, t) \in K$ , we are given a set of **taboo nodes**  $T_{s,t} \subseteq V \setminus \{s, t\}$ , which must not be contained in the routing path of commodity  $(s, t)$ .*

The concept of node taboos extends straightforward to link- or arc-taboos.

These parameters restrict the set of admissible routing paths for each commodity.

**Notation 9.15** For each commodity  $(s, t) \in K$ , an  $(s, t)$ -path in  $D$  is said to be **admissible** if it contains at most  $\ell_{s,t}$  arcs and none of the nodes in  $T_{s,t}$ . In the remainder of this chapter, we denote the set of all admissible  $(s, t)$ -paths by

$$\mathcal{P}_a(s, t) := \{ P \in \mathcal{P}(s, t) : |P \cap A| \leq \ell_{s,t} \text{ and } P \cap T_{s,t} = \emptyset \}$$

and let  $\mathcal{P}_a(K) := \bigcup_{(s,t) \in K} \mathcal{P}_a(s, t)$ .

According to these restrictions, we extend the notion of valid metrics and unsplittable shortest path routings introduced in Chapter 3 as follows.

**Notation 9.16** A routing metric  $\lambda \in \mathbb{R}_+^A$  is said to be **admissible** for the commodity set  $K \subseteq V \times V$  if

- (i) it is valid, i.e., the shortest  $(s, t)$ -path  $P_{(s,t)}^*(\lambda)$  is uniquely determined for each commodity  $(s, t) \in K$ ,
- (ii) it is symmetric, i.e.,  $\lambda_{(u,v)} = \lambda_{(v,u)}$  for all  $uv \in E$ , and
- (iii) for each commodity  $(s, t) \in K$ , the induced shortest path  $P_{(s,t)}^*(\lambda)$  is admissible.

**Notation 9.17**

Given an admissible metric  $\lambda \in \mathbb{R}_+^A$  for the commodity set  $K$ , the path set

$$S(\lambda) := \{ P_{(s,t)}^*(\lambda) : (s, t) \in K \}$$

is an **admissible unsplittable shortest path routing (USPR)** (induced by  $\lambda$ ) for the commodity set  $K$ .

In practice, we seek for an admissible metric  $\lambda$  with integer arc lengths that fit into the data format of the OSPF routing protocol, i.e., with values  $\lambda_a \in \{1, \dots, 65535\}$  for each  $a \in A$ . In the main planning process, however, this restriction can be safely ignored. For real-world size networks, any admissible metric can be turned into an equivalent admissible metric with small enough integer values using the methods discussed in Chapter 4.

Incorporating these additional restrictions into the basic path routing formulation for unsplittable shortest path routings introduced in Chapter 7 leads to the following integer linear programming characterization of admis-



sible unsplittable shortest path routings:

$$\sum_{P \in S} x_P \leq |S| - 1 \quad \forall S \in \mathcal{C}_{USPS} \cap \mathcal{P}_a(K) \quad (9.5)$$

$$\sum_{P \in \mathcal{P}_a(s,t)} x_P = 1 \quad \forall (s, t) \in K \quad (9.6)$$

$$x_P = x_{\text{reverse}(P)} \quad \forall P \in \mathcal{P}_a(K) \quad (9.7)$$

$$0 \leq x_P \leq 1 \quad \forall P \in \mathcal{P}_a(K)$$

$$x_P \in \mathbb{Z}_+ \quad \forall P \in \mathcal{P}_a(K)$$

It follows immediately from Theorem 7.3 that the vectors  $x \in \{0, 1\}^{\mathcal{P}_a(K)}$  that satisfy (9.5)–(9.7) correspond exactly to the admissible unsplittable shortest path routings for the commodity set  $K$ , i.e.,  $S(x) := \{P : x_P = 1\}$  is an admissible USPR for each  $x \in \{0, 1\}^{\mathcal{P}_a(K)}$  satisfying (9.5)–(9.7), and  $x = \chi(S)$  satisfies (9.5)–(9.7) for each admissible USPR  $S \subseteq \mathcal{P}_a(K)$ . Of course, we perform the variable substitutions corresponding to the symmetry constraints (9.7) instead of adding two binary variables with an equality constraints in the implementation of this sub-model.

Incorporating the additional routing restrictions into the arc routing formulation introduced in Chapter 7 leads directly to the following integer linear programming characterization of admissible unsplittable shortest path routings:

$$\sum_{a \in \delta^-(v)} y_a^{s,t} - \sum_{a \in \delta^+(v)} y_a^{s,t} = \begin{cases} -1, & \forall (s, t) \in K, v = s \\ 1, & \forall (s, t) \in K, v = t \\ 0, & \forall (s, t) \in K, v \in V \setminus \{s, t\} \end{cases} \quad (9.8)$$

$$\sum_{((s,t),a) \in R} y_a^{s,t} \leq |R| - 1 \quad \forall R \subseteq K \times A \text{ with} \quad (9.9)$$

$$(i) \quad \mathcal{F}(R) \in \mathcal{C}_{USPF}$$

$$(ii) \quad |\mathcal{F}(R)| = |R|$$

$$\sum_{a \in A} y_a^{s,t} \leq \ell_{s,t} \quad \forall (s, t) \in K \quad (9.10)$$

$$y_{(v,w)}^{s,t} = y_{(w,v)}^{t,s} \quad \forall (s, t) \in K, (v, w) \in A \quad (9.11)$$

$$y_{(v,w)}^{s,t} = 0 \quad \forall (s, t) \in K, (v, w) \in A \text{ with} \quad (9.12)$$

$$v \in T_{s,t} \text{ or } w \in T_{s,t}$$

$$0 \leq y_a^{s,t} \leq 1 \quad \forall (s, t) \in K, a \in A$$

$$y_a^{s,t} \in \mathbb{Z}_+ \quad \forall (s, t) \in K, a \in A$$

With Theorem 7.7 it is again easy to see that the vectors  $y \in \{0, 1\}^{K \times A}$  that satisfy (9.8)–(9.12) correspond to admissible unsplittable shortest path routings for the commodity set  $K$  and vice versa.

Note that the symmetry constraints (9.11) and the node taboo constraints (9.12) appear only for notational convenience in this routing model. In our

implementation, we of course perform the corresponding substitutions and omit the fixed variables.

#### 9.1.4 Load Groups

One of the main objectives of traffic engineering in IP networks is to minimize the maximum link congestion. In practice, however, it is often very useful to consider not only the maximum link congestion over all links in the network, but to classify the links into several groups and to consider the congestion for each group of links independently.

The topologies and the traffic demands in real networks often force the same set of links to be at the top-most load level (or even imply that the maximum congestion is attained on one particular link) no matter which routing metric is used. Minimizing the maximum congestion over all links of the network would lead to a routing that minimizes and balances the load on the one or few bottleneck links, but that disregards further potential improvements in the rest of the network. Connections to common Internet exchange points or to heavily used servers typically are such bottleneck links. If these special links are known in advance and their bandwidth cannot be increased, it is reasonable to consider them independent from the normal links.

In our traffic engineering model we allow to distinguish between arbitrarily many groups of links.

**Notation 9.18** *We are given a partition of the arc set  $A$  into **load groups**  $A_j$ ,  $j \in \mathcal{J}$ , i.e.,  $\bigcup_{j \in \mathcal{J}} A_j = A$  and  $A_k \cap A_l = \emptyset$  for all  $k, l \in \mathcal{J}$  with  $k \neq l$ . Each load group  $A_j$ ,  $j \in \mathcal{J}$ , has an associated non-negative objective coefficient  $\alpha_j \in \mathbb{Z}_+$ .*

Load groups provide the flexibility to handle the ‘special’ links known a-priori and the ‘normal’ links independent from each other. For each load group, we consider an individual value for the maximum congestion over all links in that group.

**Notation 9.19** *Let  $\lambda \in \mathbb{R}_+^A$  be an admissible metric for the commodity set  $K$  and  $u \in \mathbb{R}_+^A$  be a vector of arc capacities. For each  $j \in \mathcal{J}$ , we denote by*

$$L_j = L_j(\lambda) := \min \{ L \in \mathbb{R}_+ : f_a(\lambda) \leq L \cdot u_a \text{ for all } a \in A_j \}$$

*the maximum **congestion** that is attained by the routing induced by  $\lambda$  on the arcs of  $A_j$ .*

The objective in the traffic engineering problem now is to minimize the linear combination of these congestion values, i.e.,

$$\min \sum_{j \in \mathcal{J}} \alpha_j L_j .$$

If the arc set  $A$  is ‘partitioned’ into only one load group  $A_1 = A$ , this objective function reduces to the uniform case considered in the MIN-CON-USPR problem. If the arc set is partitioned into two or more load groups, we have to avoid that the arcs of one load group are overloaded in order to reduce the congestion on the arcs of another group. In our traffic engineering model, we therefore require that, for each load group  $j \in \mathcal{J}$ , the maximum congestion value  $L_j$  is between 0 and 1. Equivalently, we can demand that  $f_a(\lambda) \leq u_a$  for all  $a \in A$ . (Other upper bounds on the maximum admissible congestion values can be transformed to 1 by scaling the routing capacities of the respective links.)

### 9.1.5 Problems and Models at a Glance

With the above notation, the minimum cost network design problem with unsplittable shortest path routing implemented in our algorithm can be summarized as follows:

ND-USPR	
<i>Instance:</i>	<p>A link graph <math>G = (V, E)</math> and, implicitly, the associated supply digraph <math>D = (V, A)</math>.</p> <p>Components and resources, i.e.,</p> <ul style="list-style-type: none"> <li>▷ set <math>C</math> of components,</li> <li>▷ sets <math>R^L</math>, <math>R^V</math>, and <math>R^G</math> of typed resources,</li> <li>▷ component installation bounds <math>\underline{z}, \bar{z} \in \mathbb{Z}_+^{C \times \mathcal{G}}</math>, and</li> <li>▷ resource values <math>\varrho_{c,g}^r \in \mathbb{Z}</math> for all <math>c \in C</math>, <math>r \in R</math>, <math>g \in \mathcal{G}</math>.</li> </ul> <p>A commodity set <math>K \subseteq V \times V</math> with</p> <ul style="list-style-type: none"> <li>▷ demand values <math>d_{s,t} \in \mathbb{Z}_+</math>,</li> <li>▷ hop limits <math>\ell_{s,t} \in \mathbb{Z}_+</math>, and</li> <li>▷ taboo nodes <math>T_{s,t} \subseteq V \setminus \{s, t\}</math>, for all <math>(s, t) \in K</math>.</li> </ul>
<i>Solution:</i>	<p>A feasible dimensioning <math>z \in \mathbb{Z}</math> and an admissible metric <math>\lambda \in \{1, \dots, 65535\}^A</math> for the commodity set <math>K</math> such that the induced routing satisfies the capacity constraints</p> $f_{(u,v)}(\lambda) \leq \sum_{c \in C} \varrho_{c,uv}^{\text{cap}} z_{c,uv} \quad \text{for all } (u, v) \in A.$
<i>Objective:</i>	$\min \sum_{g \in \mathcal{G}} \sum_{c \in C} \varrho_{c,g}^{\text{cost}} z_{c,g}$

This problem captures not only the initial network design step, but also the re-dimensioning of an existing network. Artificial increment and decrement components and corresponding resources can be used to model the installation of additional hardware and the removal of existing hardware. Restrictions on the number of changes or on the expenses induced by the

reconfiguration can be easily formulated with these components in the dimensioning sub-model.

With the models for the dimensioning and routing subproblems presented above, we immediately obtain two integer linear programming formulations for ND-USPR. For the version with path routing variables, the corresponding arc capacity constraints are

$$\sum_{\substack{(s,t) \in K, \\ P \in \mathcal{P}_a(s,t): a \in P}} d_{s,t} x_P \leq \sum_{c \in C} \varrho_{c,vw}^{\text{cap}} z_{c,vw} \quad \forall (v,w) \in A. \quad (9.13)$$

An integer linear programming formulation for ND-USPR (ignoring the boundedness of the routing metric  $\lambda$ ) with path routing variables then is

$$\begin{aligned} \min \left\{ \sum_{g \in \mathcal{G}} \sum_{c \in C} \varrho_{c,g}^{\text{cost}} z_{c,g} : (x,z) \in \{0,1\}^{\mathcal{P}_a(K)} \times \mathbb{Z}_+^{C \times \mathcal{G}} \quad (\text{ND-PATH}) \right. \\ \left. (x,z) \text{ satisfies (9.1)–(9.7) and (9.13)} \right\}. \end{aligned}$$

For the version with arc routing variables, the capacity constraints can be written as

$$\sum_{(s,t) \in K} d_{s,t} y_a^{s,t} \leq \sum_{c \in C} \varrho_{c,vw}^{\text{cap}} z_{c,vw} \quad \forall (v,w) \in A. \quad (9.14)$$

A complete integer linear programming formulation (again, ignoring the boundedness of  $\lambda$ ) with arc routing variables is given as follows:

$$\begin{aligned} \min \left\{ \sum_{g \in \mathcal{G}} \sum_{c \in C} \varrho_{c,g}^{\text{cost}} z_{c,g} : (y,z) \in \{0,1\}^{K \times A} \times \mathbb{Z}_+^{C \times \mathcal{G}} \quad (\text{ND-ARC}) \right. \\ \left. (y,z) \text{ satisfies (9.1)–(9.4), (9.8)–(9.12), and (9.14)} \right\}. \end{aligned}$$

In traffic engineering, the goal is minimize the congestion in the network. Neither the topology of the existing network nor its hardware configuration may be changed in this short-term task. Hence, it is not necessary to consider the undirected link graph and the components and resources describing the network's fixed hardware configuration. Instead, we assume that the directed supply digraph  $D = (V, A)$  and the available routing capacities  $u \in \mathbb{Z}_+^A$  are given explicitly. The traffic engineering problem then can be formalized as follows:

TE-USPR	
<i>Instance:</i>	<p>A directed graph <math>D = (V, A)</math> with positive arc capacities <math>u_a \in \mathbb{Z}_+</math>, <math>u_a &gt; 0</math> for all <math>a \in A</math>.</p> <p>A commodity set <math>K \subseteq V \times V</math> with</p> <ul style="list-style-type: none"> <li>▷ demand values <math>d_{s,t} \in \mathbb{Z}_+</math>,</li> <li>▷ hop limits <math>\ell_{s,t} \in \mathbb{Z}_+</math>, and</li> <li>▷ taboo nodes <math>T_{s,t} \subseteq V \setminus \{s, t\}</math>, for all <math>(s, t) \in K</math>.</li> </ul> <p>Load groups <math>A_j</math>, <math>j \in \mathcal{J}</math>, with associated objective coefficients <math>\alpha_j \in \mathbb{Z}_+</math>.</p>
<i>Solution:</i>	An admissible metric $\lambda \in \{1, \dots, 65535\}^A$ for the commodity set $K$ such that the induced routing satisfies the capacity constraints $f_a(\lambda) \leq u_a$ for all $a \in A$ .
<i>Objective:</i>	<p><math>\min \sum_{j \in \mathcal{J}} \alpha_j L_j</math>, where</p> <p><math>L_j := \min \{L \in \mathbb{R}_+ : f_a(\lambda) \leq L u_a \text{ for all } a \in A_j\}</math>.</p>

Integer programming models for TE-USPR based on the models of the routing subproblems discussed above again are obtained straightforward. For the version with path routing variables, the corresponding capacity constraints are

$$\sum_{\substack{(s,t) \in K, \\ P \in \mathcal{P}_a(s,t): a \in P}} d_{s,t} x_P \leq u_a L_j \quad \forall j \in \mathcal{J}, a \in A_j. \quad (9.15)$$

A complete mixed-integer linear programming formulation for TE-USPR (ignoring the boundedness of the routing metric  $\lambda$ ) with path routing variables is

$$\begin{aligned} \min \Big\{ & \sum_{j \in \mathcal{J}} \alpha_j L_j : (x, L) \in \{0, 1\}^{\mathcal{P}_a(K)} \times \mathbb{R}_+^{\mathcal{J}} \\ & (x, L) \text{ satisfies (9.5)–(9.7) and (9.15)} \\ & 0 \leq L_j \leq 1 \quad \forall j \in \mathcal{J} \Big\}. \end{aligned} \quad (\text{TE-PATH})$$

The capacity constraints for the formulation of the traffic engineering problem TE-USPR with arc routing variables finally are

$$\sum_{(s,t) \in K} d_{s,t} y_a^{s,t} \leq u_a L_j \quad \forall j \in \mathcal{J}, a \in A_j, \quad (9.16)$$

leading to the following mixed-integer linear programming formulation of TE-USPR (ignoring the  $\lambda$ -bounds):

$$\begin{aligned} \min \quad & \left\{ \sum_{j \in \mathcal{J}} \alpha_j L_j : (y, L) \in \{0, 1\}^{K \times A} \times \mathbb{R}_+^{\mathcal{J}} \right. \\ & (y, L) \text{ satisfies (9.8)–(9.12) and (9.16)} \\ & \left. 0 \leq L_j \leq 1 \quad \forall j \in \mathcal{J} \right\}. \end{aligned} \quad (\text{TE-ARC})$$

We implemented a solution algorithm for all four formulations.

### 9.1.6 Remarks

In our project with the DFN-Verein we also encountered a combined traffic engineering and network reconfiguration problem, where modifying the existing network dimensioning within some given limits in order to reduce the network congestion as much as possible was allowed. In this problem version, both the undirected link graph with the components and resources describing the possible network reconfigurations as well as a set of load groups with associated objective coefficients are given. The task is to find a feasible dimensioning and an admissible routing metric, such that the linear combination of the load group congestion values is minimized. The side constraints that limit the number of changes to the current network configuration or the reconfiguration cost have been formulated in the dimensioning sub-model again.

Furthermore, an important issue in the project with the DFN-Verein was to plan the network in such a way, that it is still operational and if some nodes or links fail. Accordingly, also the routing metric in should be chosen such that the congestion remains as low as possible in such failure situations. In the real problems, we therefore also considered a set of so-called operating states representing typical or crucial failure scenarios. The concepts, models, and algorithms discussed throughout this thesis generalize (almost) straightforward to the respective problem versions with multiple network operating states. However, network survivability and fault tolerant routing are beyond the scope of this thesis, so we do not consider this part of the problems and of our implementation here. In [36], an earlier version of the currently implemented integer programming formulation with arc routing variables for each considered operating state is presented. Orłowski [148] and Fadejeva [83] discuss several issues concerning the integer programming formulations for closely related survivable network design problems with unsplittable flow routing, and some of their results merged into the current implementation of our algorithms for survivable unsplittable shortest path routing.

## 9.2 Implementation

In the following, we give a high-level description the algorithm that was implemented to solve the problems ND-USPR and TE-USPR.

The algorithm follows the general approach discussed in Section 7.2: In the first phase, we solve an integer linear programming formulation of the problem to find an optimal set of routing paths and, for the network design problem ND-USPR, also the corresponding network dimensioning. For each problem, either the corresponding path routing formulation (ND-PATH) or (TE-PATH) or the corresponding arc routing formulation (ND-ARC) or (TE-ARC) may be used. In the following, we refer to the chosen formulation as the Master problem. The Master problem is solved with the usual branch-and-price-and-cut or branch-and-cut algorithm. We use specialized branching rules, cutting planes, and heuristics, which are described below.

After the Master problem is solved (or the time limit is exceeded), the algorithm enters the post-processing phase. Using the algorithms presented in Chapter 4, it then computes a compatible routing metric for the best integer solution found for the Master formulation.

The algorithm is implemented in the c++ programming language as part of the DISCNET network optimization library [9]. The data structures and algorithms are based on the standard c++ library and LEDA 4.1 [132]. The linear programs arising during the solution process are solved by CPLEX [64]. CPLEX is also used to solve the integer linear programming problems that arise as sub-problems in our heuristics. The branch-and-price-and-cut framework for the Master problem, including the management of the branch-and-bound tree and the control of the pricing and separation algorithms, is part of DISCNET.

### 9.2.1 Initial Heuristics

We use some basic heuristic methods to start the branch-and-bound algorithm with a good initial solution.

#### Random routing metrics

The first heuristic simply generates a set of random routing metrics. Optionally, these metrics may be biased on the geographic link lengths or on unit lengths, which proved to be useful for instances with path length restrictions. Node taboos in the given problem are accounted for by increasing the lengths of the corresponding adjacent links. For each of these candidate metrics, the heuristic then computes the induced routing paths and traffic flows, applying a perturbation to ensure uniqueness if necessary. Metrics that induce non-admissible routing paths or whose induced flows exceed the available or maximally installable capacity are simply discarded.

For the traffic engineering problem TE-USPR, each of the remaining metrics defines a feasible solution.

For the network design problem ND-USPR, an auxiliary integer linear program is solved to find the cheapest feasible dimensioning that provides sufficient routing capacities for the induced flows. If there exists such a dimensioning, then the routing metric and this dimensioning together define a feasible solution. Otherwise the metric is discarded. For a candidate metric  $\lambda \in \mathbb{R}_+^A$ , the corresponding auxiliary integer linear program is

$$\min \left\{ \sum_{g \in \mathcal{G}} \sum_{c \in C} \varrho_{c,g}^{\text{cost}} z_{c,g} : z \in \mathbb{Z}_+^{C \times \mathcal{G}} \right. \\ \left. \begin{array}{l} z \text{ satisfies (9.1), (9.2), (9.3), and (9.4)} \\ \sum_{c \in C} \varrho_{c,vw}^{\text{cap}} z_{c,vw} \geq f_{(v,w)}(\lambda) \quad \forall (v,w) \in A \end{array} \right\}.$$

In principle these auxiliary integer programs are  $\mathcal{NP}$ -hard. However, they typically can be solved very fast in practice. In our heuristics, we limit the CPU-time for the solution (1 second per default) and choose a parameter setting instructing CPLEX to emphasize the construction of feasible solutions.

### Lagrangian algorithm

Our second initial ‘heuristic’ is the Lagrangian solution approach presented in [34]. Instead of using randomly generated link metrics, this approach iteratively updates the candidate metric based on the violation of the (relaxed) capacity constraints. Primal solutions are computed from the candidate metrics exactly as in the random weights heuristic above. As a positive side-effect, this approach also computes a lower bound on the optimal solution value. A more detailed description of this algorithm can be found in [34].

We want to point out that a good initial solution is extremely important for the performance of our solution approach. For the traffic engineering problem TE-USPR, it not only provides a cut-off value for the branch-and-bound procedure, it also helps to generate stronger cuts in the sequel, see Sections 9.2.3 and 9.2.7 below.

### 9.2.2 LP Initialization

The initial Master LP contains all variables and constraints of the corresponding unsplittable flow formulation. For the network design formulations, all component variables  $z_{c,g}$  and all resource constraints (9.1)–(9.4) are contained in the initial LP. For the traffic engineering formulations, all congestion variables  $L_j$ ,  $j \in \mathcal{J}$ , occur in the initial LP. Also, all of the respective capacity constraints (9.13), (9.14), (9.15), of (9.16) are contained in the initial LP.



Furthermore, the initial LPs of the arc routing formulations (ND-ARC) and (TE-ARC) involve (in principle) all arc routing variables, all flow conservation constraints, and all hop-length restrictions. However, only one set of arc routing variables is created for each pair of anti-parallel commodities. The variables for the other commodity are expressed in terms of those of the first commodity via the variable substitutions defined by the routing symmetry constraints (9.11). Naturally, the variables  $y_{(v,w)}^{s,t}$  with  $v \in T_{s,t}$  or  $w \in T_{s,t}$  and the corresponding taboo constraints (9.12) are omitted. Furthermore, arc routing variables  $y_{(v,w)}^{s,t}$  with  $t = v$ , with  $s = w$ , or with  $\text{dist}(s, v) + \text{dist}(w, t) \geq \ell_{s,t}$  are omitted. They obviously must attain the value 0 in any feasible solution.

For the path routing formulations (ND-PATH) and (TE-PATH), the initial LP contains all demand constraints (9.6) and the path routing variables of some initial paths. The set of these paths can be generated by (a combination of) the following methods:

- (1) Enumerate all admissible paths for each commodity  $(s, t) \in K$ .
- (2) Compute the  $k$ -shortest admissible paths for each commodity  $(s, t) \in K$  and some given  $k \in \mathbb{Z}_+$ .
- (3) Compute a maximum number of arc- or node-disjoint admissible  $(s, t)$ -paths for each commodity  $(s, t) \in K$ .
- (4) Generate a set of random routing metrics (again, these may be biased) and use all those induced routing paths that are admissible.

The USPS or USPF circuit inequalities are not contained in the initial Master LP.

### 9.2.3 Inequalities

Our algorithm separates several types of cuts and original model inequalities at the nodes of the branch-and-bound tree.

#### Metric-type inequalities

For the network design formulations (ND-ARC) and (TE-ARC), strengthened metric inequalities [5], band inequalities [66, 67], and superadditive metric inequalities are separated at the root of the branch-and-bound tree. The basic form of superadditive metric inequalities was described in Section 8.2.

The separation problem for each of these inequalities is  $\mathcal{NP}$ -hard. In our algorithm, we solve these problems heuristically for the cut-metrics given by single-node, two-node, and three-node cuts in underlying graph. Given such a graph metric, the (only) associated strengthened metric inequality is

obtained by standard integer routing and the most violated band inequality is computed using a dynamic programming algorithm [190]. In order to identify possibly violated superadditive metric inequalities, a small number of step functions  $\Gamma_c(x) = \lfloor cx \rfloor$ ,  $c \in \mathbb{R}_+$ , is applied for every given metric, c.f. Section 8.2. Which values of  $c$  are actually tried depends on the installable capacities on the links across the given cut.

Optionally, all metric-based inequalities can be separated also at deeper nodes in the branch and bound tree. Furthermore, additional base metrics can be generated during the algorithm via the standard separation techniques for classical metric inequalities or via cut enumeration in an auxiliary graph, that is obtained by heuristically shrinking the underlying link graph based on the tightness of the capacity constraints in the current LP solution. However, both options degraded the overall performance of the algorithm and are turned off per default.

### Cover and GUB-cover inequalities

In the two network design formulations (ND-PATH) and (ND-ARC), the resource constraints (9.2), (9.3), and (9.4) typically define various standard 0/1-knapsacks and GUB-knapsacks on the binary component variables  $z_{c,g}$ ,  $c \in C$  and  $g \in \mathcal{G}$ . For these knapsacks, our algorithm separates the corresponding 0/1-knapsack cover inequalities [13, 152, 111, 192] and the GUB-cover inequalities [193] at all nodes of the branch-and-bound tree. The respective separation problems are solved exactly using the obvious dynamic programming algorithms.

### Induced cover inequalities

Appropriate versions of induced cover inequalities for the precedence constrained knapsacks defined by the arc capacity constraints are separated for each of the four formulations at each node of the branch-and-bound tree. A basic version of these inequalities for a single binary capacity variable and path routing variables was discussed in Section 8.3.

For the four formulations considered here, the underlying precedence constrained knapsacks can be roughly described as follows. The basic set of items in the knapsack for arc  $a \in A$  corresponds to the arc routing variables  $y_a^{(s,t)}$ ,  $(s,t) \in K$ , or to the path routing variables  $x_P$ ,  $P \in \mathcal{P}$  with  $a \in P$ . The sizes of these items are given by the respective commodities' demand values and the precedences among them are given by the implications of the Bellman property.

For the traffic engineering formulations (TE-PATH) and (TE-ARC), the capacity of the precedence constrained knapsack on arc  $a \in A$  is  $\bar{L}_j u_a$ , where  $j \in \mathcal{J}$  is the load group with  $a \in A_j$  and  $\bar{L}_j$  is the currently best known upper bound for the maximum congestion in load group  $j$  in an optimal solution,

c.f. Section 9.2.7 below.

For the network design formulations (ND-PATH) or (ND-ARC), the component installation variables  $z_{c,vw}$ ,  $c \in C$ , on the underlying link  $vw$ ,  $(v, w) = a$ , are considered as binary or general integer items also. The sizes of these items are the negative values of the provided routing capacities, and the precedences among them are derived from GUB or ordering constraints for the corresponding component installation variables. The knapsack capacity is zero for the network design formulations (ND-PATH) or (ND-ARC).

For all four variants, the separation problem for induced cover inequalities is  $\mathcal{NP}$ -hard in general. Our algorithm employs a simple greedy heuristic to find violated induced covers. Starting with an artificial minimal item, it maintains a set of items that is closed under the precedence relations. In each iteration, it adds an item to this set. If the total size of the items in the set exceeds the capacity of the knapsack, the induced cover inequality given by the (w.r.t. the precedence relations) maximal elements of that set is evaluated. If this inequality is violated, the heuristic returns. Otherwise it continues to add items to the set, but now in such a way that the number of maximal elements of the set decreases. Every time this number reduces, the corresponding induced cover inequality is evaluated. The heuristic stops if clearly no violated induced cover inequality can be generated by adding further items. Several strategies of adding item in the first and in the second phase have been implemented. In a post-processing loop, it finally tries to reduce the generated induced covers to minimal induced covers. Lifting of the resulting basic induced cover inequalities is not performed.

### Circuit inequalities

Violated USPS circuit inequalities (9.5) for the path routing formulations (ND-PATH) and (TE-PATH) or violated USPF circuit inequalities (9.9) for the arc routing formulations (ND-ARC) and (TE-ARC) may be separated at all nodes of the branch-and-bound tree. The subprograms to find irreducible non-USPSs or irreducible non-USPFs, however, are rather time consuming and accept only path sets and forwardings as input. Therefore, attempts to separate circuit inequalities are made only if the Master LP solution is almost integer and if no other violated inequality could be found. Besides, the separation routines are also called indirectly via one of the node heuristics at the branch-and-bound nodes of depth  $2^k$ ,  $k \in \mathbb{Z}_+$ , see Section 9.2.6.

The implementation of the separation routines in our algorithm is straightforward. Given a solution of the Master LP, we first determine the path set  $S$  or the forwarding  $F$  corresponding to the integer and near-integer routing variables ( $\geq 0.8$  per default). For these sets, we then apply algorithm Greedy-Non-USPS shown on page 83 or algorithm Greedy-Non-USPF shown on page 106 to find an irreducible non-USPS  $R \subseteq S$  or an irreducible non-USPF  $E \subseteq F$ , respectively. If such sets are found, the corresponding USPS

or USPF circuit inequality is evaluated and, if violated, added to the Master LP. If the given primal solution is integer, then this approach solves the separation problem exact. Otherwise it is only a heuristic.

The linear system (4.3)–(4.4) in algorithm **Greedy-Non-USPS** is solved as described in Chapter 4, using a cutting-plane algorithm that iteratively adds violated inequalities of type (4.3). The separation problem for these inequalities is solved with an improved version of algorithm **CheckTwoShortestPaths**, which computes at most  $|V|$  many shortest path trees in each iteration of the cutting plane algorithm. The linear system (4.11)–(4.14) in algorithm **Greedy-Non-USPF** is solved via a single call to CPLEX.

### Other unsplittable shortest path routing constraints

Violated out-degree inequalities (8.7), in-degree inequalities (8.8), or sub-path inequalities (8.9), (8.10), and (8.11) for the arc routing formulations (ND-ARC) or (TE-ARC) are separated at all nodes of the branch-and-bound tree. The respective separation problems are solved exactly using ad-hoc enumeration methods. Even though there is only a polynomial number of these inequalities, we generate them only if violated to keep the size of the Master LP as small as possible.

Finally, USPS clique inequalities (8.2) for the path routing formulations (ND-PATH) and (TE-PATH) are separated at all nodes of the branch-and-bound tree. For this, our algorithm maintains a conflict graph, whose nodes represent the paths in the LP and whose edges represent the pairwise (Bellman-) conflicts among the paths.

The separation problem for the USPS clique inequalities reduces to the problem of finding a maximum weight clique in this graph, where the weights are given by the values of the path routing variables  $x_P$ ,  $P \in \mathcal{P}$ , in the current Master LP solution. In general, this problem is  $\mathcal{NP}$ -hard. However, there are numerous efficient heuristics and exact solution methods for finding maximum weight cliques in general graphs that can be applied. Bomze et al. [38] provide a very comprehensive overview of such methods. In our algorithm, we use the sequential Greedy heuristic described by Borndörfer and Kormos [40], which proved to be very efficient. Given the size of the conflict graphs we encounter in our application, computing an optimal maximum weight clique using the branch-and-bound algorithm proposed in [40] or some other method is way out of the question.

#### 9.2.4 Pricing of Path Variables

If the Master problem is one of the path routing formulations (ND-PATH) or (TE-PATH) and the path variables have not been completely enumerated initially, our algorithm adds path variables with negative reduced costs at each node of the branch-and-bound search tree.

The corresponding pricing problem is solved using the  $k$ -shortest paths approach described in Section 7.5: For each commodity  $(s, t) \in K$ , let  $k_{s,t}$  be the number of  $(s, t)$ -path variables in the current restricted LP and let  $Z_{s,t} \subset A$  be the set of arcs for which the flow of commodity was fixed to 0 by previous branching decisions. The hop limit for admissible  $(s, t)$ -paths is  $\ell_{s,t}$ . For each commodity  $(s, t) \in K$  whose routing path is not yet entirely fixed by the branching decisions and whose set of admissible routing paths has not been fully added to the LP, our algorithm computes the  $\ell_{s,t}$ -bounded  $(k_{s,t}+1)$ -shortest  $(s, t)$ -paths in the digraph  $D - T_{s,t} - Z_{s,t}$  with respect to the lengths given by the dual variables of the capacity constraints. The paths are computed iteratively in order of increasing length (w.r.t. the dual variables), using a modified version of Yen's  $k$ -shortest path algorithm [196, 12]. The procedure stops as soon as a path  $P$  is found whose corresponding routing variable  $x_P$  is not contained in the restricted LP. If this path  $P$  has negative reduced cost, then it is added to the LP. Otherwise, if  $P$  has non-negative reduced cost or if  $k_{s,t} + 1$  paths have been considered, none of the admissible  $(s, t)$ -paths has negative reduced cost.

Note that our algorithm does not lift the new variables into existing circuit, clique, or induced cover inequalities. The pricing problem would become strongly  $\mathcal{NP}$ -hard (even with respect to the size of the restricted LP), if immediate lifting is considered. (In practice, however, this no-lifting approach is not very efficient, as we will see in our computational results.)

### 9.2.5 Branching

#### General branching scheme

For the branching on component variables and arc routing variables, our algorithm uses standard branching on the variable bound.

For the branching on path routing variables, it uses the set-based branching scheme proposed by Barnhart et al. [18] for unsplittable flow problems. Given a commodity that is routed fractionally in the current LP solution, the algorithm first determines a vertex  $v$  where the commodity's flow splits, i.e., where at least two of the outgoing arcs carry a positive fraction of the commodity's flow. Then it partitions the set of all outgoing arcs into two disjoint subsets  $A_1 \cup A_2 = \delta^+(v)$ ,  $A_1 \cap A_2 = \emptyset$ , such that both of these subsets carry a positive fraction of the commodity's flow. Finally, it creates two child nodes in the branch-and-bound tree. The first child node is defined by the fixing  $x_P = 0$  for all  $P \in \mathcal{P}(s, t) \cap A_1$  (also for those paths not even in the LP), the second child is given by the fixing  $x_P = 0$  for all  $P \in \mathcal{P}(s, t) \cap A_2$ .

### Branch node selection

The next branch-and-bound node to explore is selected by the following strategy. For a given number of iterations (32 by default) the algorithm chooses the node with the best dual bound. Then, it ‘dives’ for a good feasible solution deep into the tree. The dive starts at a best-dual-bound node and then always chooses the child node whose arc routing variable or component variable was rounded up by the last branch. When branching on sets of path routing variables, it chooses the child where the biggest of the affected path variables was not fixed to zero. If a new feasible solution is found, the algorithm switches back to the best-dual-bound strategy. If it ends up in an infeasible or too costly node of the branch-and-bound tree, it backtracks. If after a fixed number of backtrack steps (5 by default) no feasible solution is found, it switches back to the best-dual-bound strategy.

### Branch variable selection

The aim of our branch variable selection policy is to perform first the ‘important’ branches, which fix the routing variables of the big demand commodities or the installation variables of expensive and high capacity components. Routing variables of small demand commodities shall be handled at the deeper nodes of the branch-and-bound tree.

For the network design formulations (ND-PATH) and (ND-ARC), our algorithm uses a strategy that alternates between branches on the routing variables and branches on the component variables.

When branching on routing variables, it always prefers the routing variables that occur in tight capacity constraints or tight induced cover inequalities. When solving a traffic engineering problem, this means that it prefers the fractional routing variables that occur in the capacity constraints where the maximum congestion for the respective load group is attained. Among the preferred routing variables, it then chooses one whose value is ‘reasonably’ fractional and whose associated demand value is big.

#### 9.2.6 Primal Heuristic

Two primal heuristics are used at the nodes of the branch-and-bound tree. Both are applied only at nodes with depth  $2^k$ ,  $k \in \mathbb{Z}_+$ , or if the ratio between the optimal LP value of the current node and the optimal LP value of its parent node exceeds a certain threshold (1.02 by default). Thus, the heuristics are applied more frequently at the top of the branch-and-bound tree than in the lower parts of the tree, and they are applied after very significant changes.

Analogous to the initial heuristics, both heuristics generate a candidate routing metric and then try to compute a feasible solution for this metric.

For the network design problem ND-USPR, this again involves the solution of an auxiliary integer program.

Our first heuristic tries several combinations of the current dual variables of the capacity constraints and (biased) random metrics in the digraph given by the current fractional arc capacity installation. As in the initial heuristics, these candidate metrics are manipulated to account for path length restrictions and node taboos in the given problem. In addition to the modifications performed by the initial heuristics, the heuristics used at the nodes of the branch-and-bound tree also increase the length of those links where the traffic flow in the current LP solution is less than 50% of the smallest still installable capacity. With this simple technique we try to avoid the use of links that very likely are not installed in the best solution of this branch-and-bound subtree.

Our second heuristic is closely related to the separation routine for USPS or USPF circuit inequalities. In a first step, it determines the path set or the forwarding that is given by the integer or near-integer ( $\leq 0.8$  per default) path or arc routing variables. Then it tries to find a compatible metric for this routing or forwarding by solving the linear programming relaxation of (ARC-IUSP) or (ARC-IUSPF).

If this linear program is solvable, its solution comprises a candidate routing metric, which is processed exactly as in the first heuristic. Otherwise, if the linear program is not solvable, the near-integer routing variables form a non-USPS or a non-USPF. In this case, the heuristic proceeds like the separation algorithm for the respective circuit inequalities, see Section 9.2.3.

### 9.2.7 Further Techniques

#### Flow costs

We observed that the LP relaxations of the Master problem are solved faster if the routing variables have a positive objective function coefficients. In our implementation, we perturb the original objective function of the Master problem by adding small flow costs.

This technique encourages the LP to route commodities via short rather than long paths. This reduces several stalling effects, such as the repeated generation of slightly violated inequalities followed by a small shift of the flows in the cutting plane phase. For the performance of the overall algorithm, this perturbation technique turned out to be very important.

#### Automatic bound strengthening

When solving the traffic engineering problem TE-USPR, we also apply some simple but effective bound strengthening for the congestion variables. For each load group  $j \in \mathcal{J}$ , we keep track of an upper bound  $\bar{L}_j$  and a lower bound  $\underline{L}_j$  for the congestion variable  $L_j$ . The upper bounds  $\bar{L}_j$  are

set to 1 initially. An initial value for the lower bounds can be computed by considering which arcs of each load groups must be used in a solution. If, for example, all arcs of a graph cut  $\delta(W)$  belong to the same load group  $A_j$ , then we immediately have  $\underline{L}_j \geq \max\{d_{v,w}/u_a : w \in W, v \in V \setminus W, a \in A_j\}$ .

Whenever a new feasible solution with objective  $L^*$  is found, we try to tighten the upper bounds by setting

$$\bar{L}_j := \min \left\{ \bar{L}_j, 1/\alpha_j (L^* - \sum_{k \in \mathcal{J} \setminus j} \alpha_k \underline{L}_k) \right\}, \quad \text{for all } j \in \mathcal{J}.$$

The upper bounds for the congestion variables  $L_j$ ,  $j \in \mathcal{J}$ , in the Master LP are updated accordingly.

This bound strengthening indirectly tightens the capacity constraints and thereby reduces the remaining search space for the branch-and-bound algorithm. This effect is amplified by the fact that the induced cover inequalities are based on these capacity constraints. The reduction of the upper bounds for the congestion variables  $L_j$ ,  $j \in \mathcal{J}$ , immediately leads to a reduction of the capacities  $\bar{L}_j u_a$  that are considered in the corresponding precedence constrained knapsack problems. The capacity reduction introduces new induced covers for these knapsacks and thus leads to more and stronger induced covers inequalities in the following separation attempts. Of course, these inequalities are valid only for the restricted problem of finding a better solution than the best known one.

### 9.2.8 Post-Processing

In the post-processing phase of our algorithm, a compatible metric for the best solution of the Master problem is determined. Using the methods presented in Chapter 4, the implementation is straightforward.

If the Master problem is formulated in terms of path routing variables, we first solve the linear relaxation of (ARC-IUSP) for the set of routing paths given by the Master problem's solution. This yields a fractional compatible metric for these paths. Then we perform the scaling and rounding steps described in algorithm **MIP-Rounding** to construct a compatible metric with integer arc lengths. If the Master problem is formulated in terms of arc routing variables, we proceed analogously. First we solve the linear relaxation of (ARC-IUSPF) for the forwarding defined by the Master problem's solution. Analogous to algorithm **MIP-Rounding**, we then round and scale the resulting fractional metric to an integer one. In both cases, we can re-use the linear programs for the inverse shortest paths or inverse shortest path forwarding problem, respectively, that have been used in the first phase of the algorithm.



## 9.3 Computational Results

We tested our algorithm on various real-world problems from the DFN-Verein and on several benchmark instances taken from the Survivable Network Design Data Library SNDlib [149]. Our main interest, of course, was to solve the problems arising in the planning of the German national research and education network as good as possible. In this section we present and discuss some selected computational results.

All computations were performed on an Intel Pentium 4EE machine with 3.2 GHz speed and 2 GB RAM running Linux 2.6.16. The LP relaxations of the Master problem and the auxiliary integer linear programs arising in the heuristics were solved by CPLEX 10.1 [64].

### 9.3.1 DFN Instances

We consider 12 network design and 6 traffic engineering problems, that originate from 6 different planning turns of the German national research network between 1999 and 2005. The bwin problems stem from the last re-planning of the B-WiN in 1999, the problem gwin1 to gwin4 come from different network expansion stages of the G-WiN, and the xwin instances arose in the first design studies of the current IP-over-optics backbone network X-WiN. For each of the 6 planning turns we have two network design and one traffic engineering problems.

The bwin and gwin network design problems are very similar. They are all defined on a complete commodity set describing the full traffic matrix and a complete or almost complete link graph on 10 or 11 nodes. These were the 10 top level backbone nodes and, in the gwin1–3 problems, an artificial node with two adjacent edges representing the two parallel ‘up-links’ to the global Internet or, in gwin4, an important node at the next lower network level. The hop limits in the bwin and gwin problems are 2 or 3, depending on the actual commodity, and neither the artificial node nor the lower level node must carry transit traffic.

In the bwin problems, one of 55 possible capacities can be chosen for each edge, but the number of chosen edges must be either 12, 13, or 14. The total network cost is given by a curious function depending on the number of installed edges, and the minimum capacity installed on these edges, and the capacity installed on each. It is described in detail in [37]. Node hardware is not considered in the bwin problems.

In the gwin problems, the dimensioning of the artificial node and its adjacent edges is fix. On each of the other links, one capacity from a given set of capacities can be installed. Depending on the specific problem instance, these sets represented various subsets of the STM-hierarchy. At the nodes, the corresponding interface cards for both routers and switches are considered. Depending on the actual instance, the total installed link capacity,

the number of links of each type, and the number of interface cards of each type can be bounded. Finally, the dimensioning must provide also enough spare ports to connect the local networks to the backbone router.

For each of the 5 basic bwin and gwin settings, a greenfield network design problem, a network reconfiguration problem, and a traffic engineering problem are considered. In the greenfield network design problem, the dimensioning (and the topology) of the network may be chosen arbitrary with respect to the side-constraints discussed above. In the reconfiguration problem, the total number of link modifications with respect to the existing network is bounded. The traffic engineering problem finally allows no changes to the configuration of the existing network. The problems bwin and gwin1 correspond to real greenfield network design problems solved for a top-to-bottom reconfiguration of the B-WiN and the initial design of the G-WiN. The reconfiguration restrictions in bwinR and gwin1R are artificial. Problems gwin2R, gwin3R, and gwin4R, on the other hand, correspond to real reconfiguration problems solved in the course of network expansions of the G-WiN. Here, the reconfiguration restrictions have been removed in the respective greenfield version gwin2, gwin3, and gwin4.

The network design problems xwin1 and xwin2 stem from the planning of the current IP-over-optics backbone and are simplified versions of the real problem. In these problems, small multiples of 10 Gbit connections can be installed between the nodes. In xwin1, connections can be installed only between nodes that are direct neighbors in the underlying dark fiber network and for some extra node pairs. The link graph of xwin2 contains additional links that correspond to optically paths in the underlying dark fiber network. At the nodes, interface cards for the IP router and for the optical equipment are considered. The 250 biggest demand values obtained from traffic accounting form the commodity set of both xwin problems. For each commodity, the hop limit is set to the length of the respective shortest path plus 5, and the artificial nodes representing external networks must not carry transit traffic. The traffic engineering problem was derived by fixing the link capacities to the solution computed in the greenfield planning problem.

Table 9.1 on page 250 presents the computational results obtained with our algorithm for the 12 DFN network design problems. The reconfiguration problems are marked with an 'R'. All path variables for the path routing formulation were enumerated initially. Note that the computations were performed with the real DFN problem data and side constraints, but not with the original cost data. The used cost functions resemble the characteristics of the original ones, but result in different cost values (and in some cases also in different optimal solutions).

The columns in this and the following tables show the following values: The columns  $|V|$ ,  $|E|$ , and  $|K|$  display the number of nodes and edges in the link graph and of directed commodities. For each integer linear programming

formulation, the column LP shows the value of the initial LP formulation. Root shows the lower bound after finishing the price-and-cut algorithm at the root node of the branch-and-bound tree. LB and UB display the values of the best lower bound and the best feasible solution at the end of the branch-and-bound algorithm, Nodes the number of branch-and-bound nodes evaluated, Gap the remaining optimality gap in %, and Time the total running time of the algorithm in seconds.

The computational results for the 6 traffic engineering problems are shown in Table 9.2 on page 251. Again, all path variables were enumerated initially. In the problems bwins and gwin4 the goal is to minimize the maximum link congestion over all arcs. In the other four problems, gwin1, gwin2, gwin3, and xwin1, the artificial up-links form a second load group, and the objective is to minimize a weighted sum of the maximum congestion over all up-links and the maximum congestion over all normal links.

To the satisfaction of our project partner, the implemented algorithm can solve almost all DFN instances to optimality within reasonable time.

The results in Table 9.1 show that the greenfield network design problems are somewhat harder to solve than the corresponding network reconfiguration problems. For the dense bwins and gwin instances, the lower bounds obtained at the root node, the number of branch-and-bound nodes, and the overall solution times are comparable for both formulations. Given the tight hop length restrictions, this is not surprising. The xwin instances are not very tightly hop-restricted, but very sparse. Here the path routing formulation yields clearly better results. The initial LP gaps are moderate for all instances and formulations. On average, roughly 50% of the initial gap are closed at the root node of the branch-and-bound tree in both formulations.

Compared to the network design problems, the traffic engineering problems can be solved very fast. One reason is that the traffic engineering problems are based on the sparse actually realized networks and not on the (nearly) complete ones considered in the network planning step. The main other reason is that our initial heuristics perform very well for traffic engineering problems, especially on such sparse graphs. Together with the automatic bound strengthening based on these solutions, this helps to separate strong induced cover inequalities and solve the problems after a few branches on the big demands' routing variables. The initial LP gap is closed basically by the combination of branching and induced cover inequalities in these tests.

### 9.3.2 Instances from the SNDlib

A second set of test instances was taken from the SNDlib [149]. The SNDlib is a library of publicly available benchmarking instances for survivable network design problems. Most of the instances in the SNDlib originate from

Name	$ V $	$ E $	$ K $	arc routing formulation							path routing formulation						
				LP	root	LB	UB	Nodes	Gap	Time	LP	root	LB	UB	Nodes	Gap	Time
bwin	10	45	90	9757	9769	9775	9924	258	1.5	10000	9757	9764	9778	9778	156	0.0	4296
bwinR	10	45	90	1332	1333	1334	1334	321	0.0	4537	1332	1333	1334	1334	94	0.0	3861
gwin1	11	47	110	6326	6833	7117	7329	24301	3.0	10000	6326	6825	7038	7399	3436	5.1	10000
gwin1R	11	47	110	6703	7597	7991	7991	1020	0.0	185	6703	7573	7991	7991	1671	0.0	861
gwin2	11	47	110	6342	6850	7266	7266	1615	0.0	281	6342	6900	7266	7266	6572	0.0	2803
gwin2R	11	47	110	7831	8742	9358	9358	890	0.0	65	7831	8835	9358	9358	689	0.0	39
gwin3	11	47	110	1440	1518	1561	1561	717	0.0	66	1439	1517	1561	1561	446	0.0	54
gwin3R	11	47	110	1501	1592	1666	1666	226	0.0	24	1501	1597	1666	1666	170	0.0	22
gwin4	11	55	110	9795	1010	1131	1131	2195	0.0	1226	9795	1009	1131	1131	2233	0.0	2466
gwin4R	11	55	110	1354	1751	1946	1946	286	0.0	67	1354	1741	1946	1946	158	0.0	117
xwin1	42	58	250	1096	1115	1145	1227	2780	7.2	10000	1109	1139	1227	1227	6328	0.0	432
xwin2	42	135	250	1102	1129	1192	1246	10816	4.5	10000	1101	1130	1184	1184	17574	0.0	1566

Table 9.1: Results DFN network design problems. (Only the first 4 digits of the values displayed.)

Name	$ V $	$ E $	$ K $	arc routing formulation							path routing formulation						
				LP	root	LB	UB	Nodes	Gap	Time	LP	root	LB	UB	Nodes	Gap	Time
bwin	10	12	90	977	989	1000	1000	6	0.0	6	977	977	1000	1000	8	0.0	8
gwin1	11	19	110	903	925	1000	1000	4	0.0	1	903	918	1000	1000	14	0.0	2
gwin2	11	27	110	890	910	1000	1000	163	0.0	21	890	899	1000	1000	222	0.0	30
gwin3	11	23	110	828	856	1000	1000	46	0.0	6	828	856	1000	1000	88	0.0	9
gwin4	11	23	110	798	1000	1000	1000	6	0.0	1	798	1000	1000	1000	2	0.0	1
xwin1	42	58	250	958	959	1000	1000	74	0.0	57	958	958	1000	1000	7724	0.0	5679

Table 9.2: Results for DFN traffic engineering problems. (Values scaled such that the larger of the two  $LB$ -values is 1000.)

real-world network dimensioning problems. Because many of the problems in the SNDlib are intrinsically undirected and provide only one demand value (of unclear direction) for each node pair, we made the demand values in all computations with SNDlib instances symmetric.

The tested network design problems contain for each link a finite set of capacities to chose from. The 5 problems dfn-bwin, dfn-gwin, dfn-xwin, diyuan, and tal use the set of link capacities and costs provided in the original SNDlib instance.<sup>1)</sup> In the remaining problems, the original capacities are interpreted as basic units that can be installed in integer combinations with multiplicities up to 10. The costs of the created capacities are the sum of the base unit costs. Node components are not considered in these instances.

The SNDlib traffic engineering instances have been constructed by setting all link capacities uniformly to a sufficiently large value.

Tables 9.3 and 9.4 on pages 253 and 254 report the computational results obtained for these instances with hop limits set to the length of the respective shortest path plus one for each commodity. For this setting, all variables in the path routing formulations were enumerated initially.

Table 9.5 on page 255 lists some results for the traffic engineering problems without hop limit. In this setting, only the 20 shortest paths were generated initially for the path routing formulation, the remaining path variables were priced out during the run of the algorithm. The columns Paths and B-Conflicts show the total number of path variables created by the algorithm (until the total time limit was exceeded) and the number of pairwise (Bellman-type) conflicts among these paths.

For the SNDlib instances with tight hop limits, our algorithm shows the same principle behavior as for the DFN instances. Except for norway, all traffic engineering problems could be solved to optimality. The running times for the two alternative formulation are comparable. As in the DFN instances, the initial LP gap of the traffic engineering problems is closed essentially within the branch-and-bound tree and not at the root.

For the norway instance one observes that only relatively few branch-and-bound nodes are explored until the time limit is exceeded. The reason is that in this instance the number of admissible routing paths is large (even for the given tight hop limit) and, more important, that there are many interdependencies among these paths. There are 4260 admissible routing paths and 2,645,994 pairwise conflicts among them. This means that our clique inequality separation sub-routine for the path routing formulation searches for maximum weight cliques in a graph with 4260 nodes and 2,645,994 edges. Even with the used heuristics this is time consuming. Furthermore, the great number of admissible routing paths leads to some stalling effects

---

<sup>1)</sup>The SNDlib problems dfn-bwin, dfn-gwin, and dfn-xwin are descendants of the original DFN instances bwin, gwin2, and xwin1. They have been simplified to fit into the formats of the SNDlib and modified to conceal the original cost data.

Problem	$ V $	$ E $	$ K $	arc routing formulation							path routing formulation						
				LP	root	LB	UB	Nodes	Gap	Time	LP	root	LB	UB	Nodes	Gap	Time
dfn-bwin	10	45	90	1795	5698	7417	7417	100	0.0	183	1795	7417	7417	7417	14	0.0	78
dfn-gwin	11	47	110	1643	2165	2944	2988	32610	1.1	10000	1643	2205	2988	2988	25633	0.0	3021
dfn-xwin	42	58	250	952	1163	1227	1227	11968	0.0	580	950	1144	1227	1227	14053	0.0	202
atlanta	15	22	210	477	6516	9570	9570	36	0.0	24	477	4100	9570	9570	101	0.0	41
di-yuan	11	42	44	3161	7129	8124	8124	124	0.0	18	3161	7189	8124	8124	133	0.0	15
pdh	11	34	48	459	1138	1304	1304	280	0.0	12	459	1168	1304	1304	178	0.0	12
polska	12	18	132	5866	7329	8555	8555	250	0.0	28	5866	7114	8555	8555	291	0.0	23
tal	24	55	396	235	1052	1078	1078	8	0.0	30	235	1078	1078	1078	1	0.0	56
norway	27	51	702	854	6276	6377	6377	6	0.0	88	854	6373	6377	6377	3	0.0	1299
newyork	16	49	240	361	1503	1815	1815	266	0.0	206	361	1295	1542	1889	92	22.5	10000
france	25	45	600	1888	1991	1997	2320	2388	16.2	10000	1888	1990	1994	2340	297	17.4	10000
nobel-us	14	21	182	797	7092	8673	8673	48442	0.0	3611	797	6967	8673	8673	45074	0.0	3320
nobel-ger	17	26	242	1154	1241	1430	1430	10225	0.0	1358	1154	1220	1399	1430	11680	2.4	10000
nobel-eu	28	41	756	5707	5948	6118	6554	2984	7.1	10000	5707	5943	6027	6630	1044	10.0	10000

Table 9.3: Results for SNDlib network design problems. Hop limits set to shortest path length plus one. (Only the first 4 digits of values displayed.)

Problem	$ V $	$ E $	$ K $	arc routing formulation							path routing formulation						
				LP	root	LB	UB	Nodes	Gap	Time	LP	root	LB	UB	Nodes	Gap	Time
dfn-bwin	10	45	90	576	659	1000	1000	338	0.0	6	576	576	1000	1000	516	0.0	109
dfn-gwin	11	47	110	992	992	1000	1000	58	0.0	1	992	992	1000	1000	236	0.0	2
dfn-xwin	42	58	250	997	997	1000	1000	97	0.0	8	997	997	1000	1000	183	0.0	5
atlanta	15	22	210	961	977	1000	1000	3	0.0	1	961	961	1000	1000	3	0.0	1
di-yuan	11	42	44	491	557	1000	1000	51	0.0	2	491	568	1000	1000	28	0.0	1
pdh	11	34	48	558	665	1000	1000	5	0.0	1	558	633	1000	1000	18	0.0	1
polska	12	18	132	991	993	1000	1000	306	0.0	5	991	991	1000	1000	624	0.0	6
tal	24	55	396	739	898	1000	1000	349	0.0	28	739	826	1000	1000	20	0.0	1199
norway	27	51	702	1000	1000	1000	1119	891	11.9	10000	1000	1000	1000	1119	504	11.9	10000
newyork	16	49	240	718	718	1000	1000	28	0.0	13	718	718	1000	1000	18	0.0	8
france	25	45	600	790	790	1000	1000	198	0.0	127	790	790	1000	1000	138	0.0	149
nobel-us	14	21	182	951	984	1000	1000	1098	0.0	25	951	984	1000	1000	2147	0.0	11
nobel-ger	17	26	242	885	885	1000	1000	59	0.0	6	885	885	1000	1000	55	0.0	25
nobel-eu	28	41	756	997	997	1000	1000	5740	0.0	3120	997	997	998	1000	1292	0.3	10000

Table 9.4: Results for SNDlib traffic engineering problems. Hop limits set to shortest path length plus one. (Values scaled such that the larger of the two  $LB$ -values is 1000.)



Problem	arc routing formulation							path routing formulation							
	LP	root	LB	UB	Nodes	Gap	Time	root	LB	UB	Nodes	Gap	Time	Paths	B-Conflicts
atlanta	765	765	1000	1000	62	0.0	18	765	765	1035	26	35.3	10000	2610	2,651,457
polska	991	991	1000	1000	1226	0.0	106	991	991	1051	139	5.9	10000	2430	2,606,248
france	879	879	1000	1135	90	13.5	10000	879	879	1352	6	53.8	10000	7392	12,549,244
nobel-us	960	960	1000	1000	145	0.0	28	960	960	1169	84	21.7	10000	7075	23,366,655
nobel-ger	885	885	1000	1000	106	0.0	252	885	885	1188	26	34.2	10000	3034	3,449,109
nobel-eu	997	997	1000	1000	160	0.0	46	997	997	1161	9	16.7	10000	9633	22,870,658

Table 9.5: Results for some SNDlib traffic engineering problems without hop limits. (Values scaled such that the larger of the two *LB*-values is 1000.)

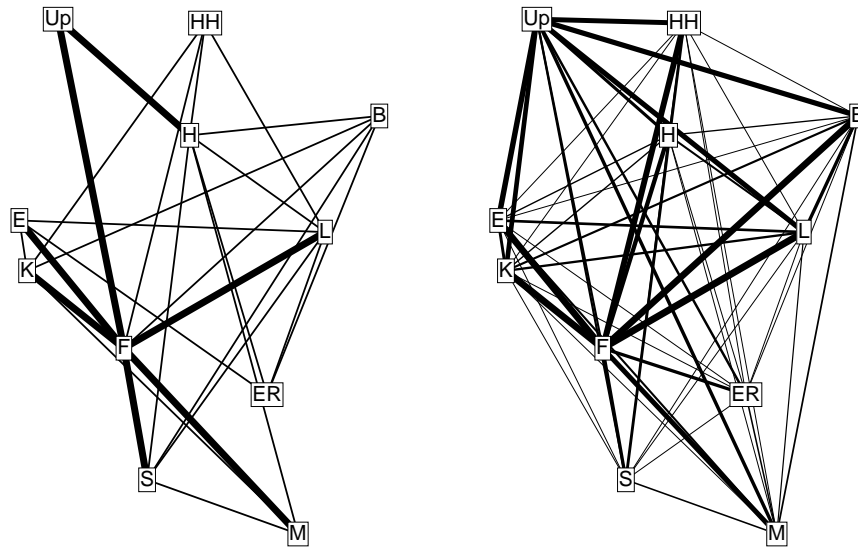
in the cutting-plane algorithms at the nodes of the branch-and-bound tree: In both formulations the LP has enough freedom to repeatedly reroute the flows in order to satisfy the inequalities that have just been added without changing the objective function.

As shown in Table 9.3, most of the small network design problems could be solved to optimality. For those problems that could not, the algorithm either does not improve on the initial solution at all or it finds a solution that is close to the best solution reported in the end very quickly. This indicates that our initial heuristics and our branching strategies and local heuristics work very well. In fact, finding good solutions quickly was one of the main goals in the design of the branching strategies. Also the dual bound improves mainly in the first few branch-and-bound nodes, where branches on the biggest and most expensive capacities and on the big demands' routing variables are performed.

Some of network design problems have significant gaps between the initial LP value and the root LP value after adding cuts. However, the largest fraction of these gaps could be closed with band inequalities and superadditive metric inequalities and was not related to the unsplittable shortest path routing nature of the problems.

For the smaller problem instances, both formulations are comparable. For the harder instances, the arc routing formulation typically outperforms the path routing formulation. Apparently, the inequalities used in the path routing formulation, which are focused on individual paths rather than arc flows, admit to much flow rerouting and thus lead to stalling effects described above. The inequalities in terms of arc routing variables seem to better 'get to the point'.

The results reported in Table 9.5 for the traffic engineering problems without hop limits corroborate this observation. While with the arc routing formulation most of the problems still can be solved reasonably well, the path routing formulation utterly fails. For the presented instances, it did neither improved the solutions provided by the initial heuristics nor the lower bound obtained by the price-and-cut algorithm in the root node. The reported numbers of generated paths and Bellman-conflicts among them describe the size of the graphs that have been considered in the separation subroutine for clique inequalities. In addition to the stalling effects described above, we also observed another kind of stalling when solving the price-and-cut problems at the nodes of the branch-and-bound tree: As new path variables are not lifted into the existing inequalities during the pricing procedure, the following separation phase often will do this lifting job and generate one of the already inequalities with the new variables lifted in. In the worst case, this iterates a couple of times and the same basic inequality is generated for several 'lifting stages'. This is obviously very inefficient and needs to be changed in order to make the path routing formulation applicable also for problems with many admissible paths.



**Figure 9.4** Topology and traffic demands of G-WiN-2.

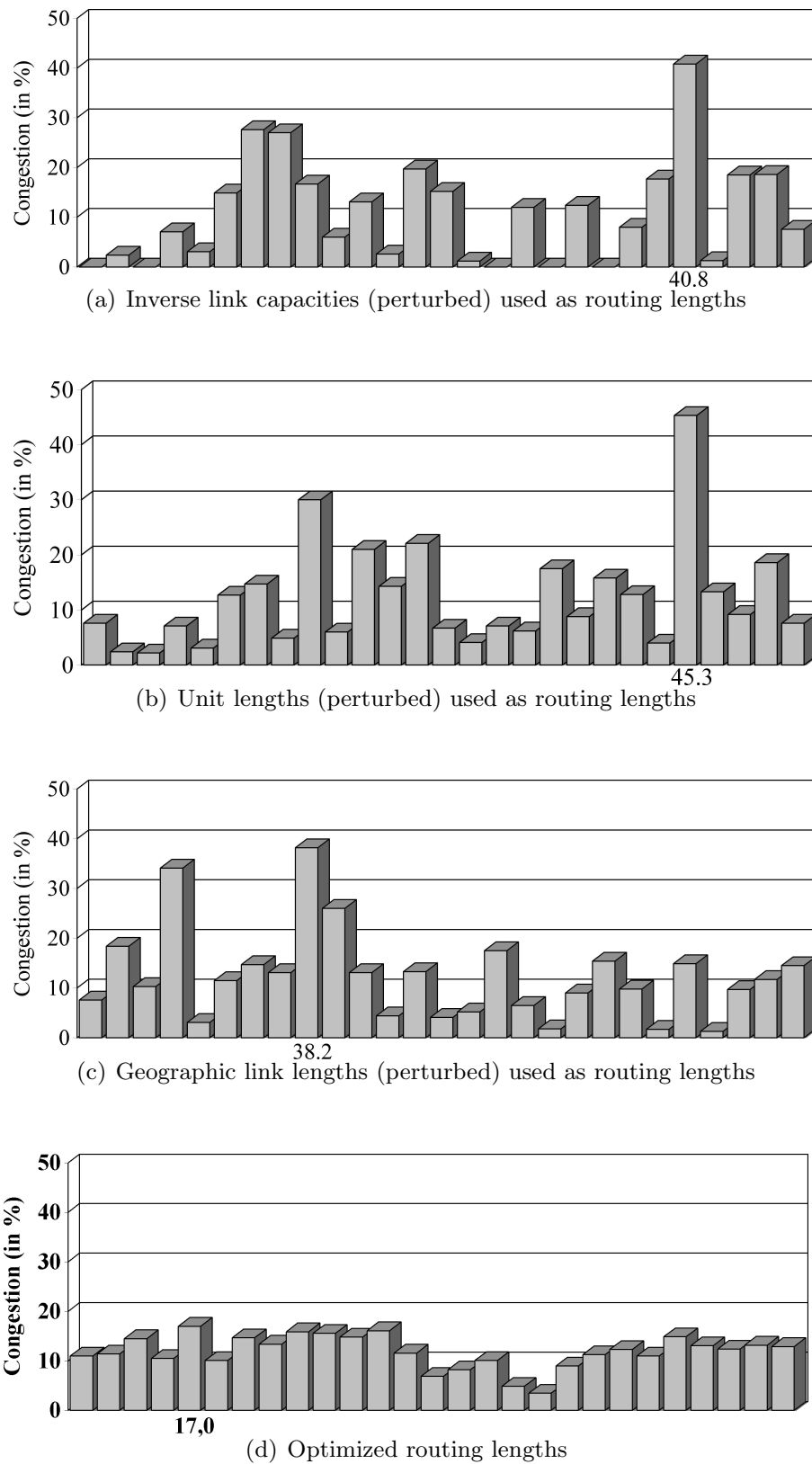
### 9.3.3 Proof of Practical Importance

We conclude this section with a small example illustrating the practical importance of our optimization algorithm for the DFN-Verein.

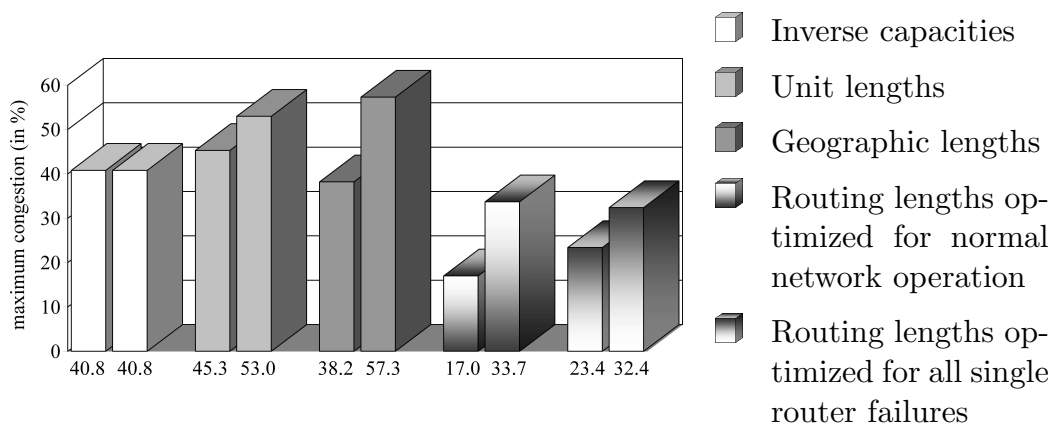
Figure 9.4 illustrates the topology and the traffic demands of the G-WiN-2 network in August 2001. The traffic engineering problem gwin2 considered above originates from these data.

Most manufacturers pre-configure their routers with a set of alternative default metric settings. Of course, these settings are made without knowledge of the network that the router will be used in, least of all of the traffic demands in that network. Nevertheless, these default settings are used in surprisingly many networks. For large and dense networks with homogeneous capacities and evenly distributed traffic demands the default setting may even perform reasonably well. For the G-WiN-2 network, however, using one of the typical default routing metrics would have been extremely bad. Figure 9.5 on the following page shows the link congestion values that result from three of the most commonly used routing metric settings and those that result from the optimal metric computed with our integer linear programming algorithm.

Obviously, the traffic is distributed more evenly over the links with the optimized routing metric. The peak congestion is not even half of that observed for three default settings. With the balanced routing, the network will be much more robust in case of unforeseen traffic changes or in case of network failures. To illustrate this, Figure 9.6 on page 259 shows not only the peak congestion in the normal network state, but also the peak congestion attained if any single router fails and the OSPF protocol is left to reroute all traffic in the residual network.



**Figure 9.5** Link congestion values for several routing metrics.



**Figure 9.6** Maximum link congestion values in normal operation and in single router failure scenarios for several routing metrics.

## 9.4 Conclusions and Future Work

The results of our computational experiments show that the proposed integer linear programming approach works well for small and medium-size problems. In particular, almost all real-world planning problems of the DFN-Verein could be solved to optimality. For those problems that could not be solved satisfactory, our algorithm provides good solutions quickly.

For the solution of problems with unevenly distributed traffic demands the integer programming approach is particularly favorable. Because its decisions are based on the end-to-end routing paths, it can force the dominant commodities explicitly to use or not to use certain paths or links and assess the effect that these alternatives have on the rest of the problem. Another advantage of using an integer programming based method is that additional technical or operational side constraints can be incorporated rather easy. This makes the algorithm very flexible and allows its application for various problem settings.

Although the implemented algorithm was able to solve the problems of primary interest satisfactory, it still has several deficiencies. The arc routing formulation works reasonably well also for larger problems, but the method used to solve the path routing formulation is inappropriate for problems with many admissible routing paths. The implemented pricing approach certainly needs a revision to make the path routing formulation applicable to larger problems. Adding inequalities that are originally based on the arc routing or forwarding variables may be beneficial. In order to solve large problems, even a pure path-reformulation of the arc routing model may be useful.

The computational results also reveal that only a small fraction of the integrality gap was closed at the root node of the branch-and-bound tree. The added cuts only became effective after a few branches. Finding stronger cuts will certainly lead to a better performance of the overall algorithm.

Other branching rules might lead to improvements as well. The currently implemented ones focus on demand values and capacities, but they disregard the interdependencies among the routing paths.

Finally, we need to remark that both the arc routing and the path routing formulation have inherently bad scaling properties. The approach of modeling each commodity's path with an individual set of binary routing variables naturally leads to big integer linear programs. In view of the computational results for the larger test instances, it might be useful to reconsider several ideas and approaches that have been suspended after some first test on the smaller instances. Using an extended formulation with forwarding variables, for example, might reduce the number of cut generation loops and thereby improve the overall solution time. For bigger problems, it might be better to use aggregated formulations, where only the forwardings and the reversely directed anti-forwardings are described with binary variables, but continuous variables are used to model the aggregated traffic flows. The big-M characteristic of these formulations may be compensated by the smaller size of the resulting integer programs. Formulations that model only a few dominant traffic demands explicitly by binary routing or forwarding variables and consider the rest of the problem as a continuous multicommodity routing finally might be an option to deal with really large problems.

# Bibliography

- [1] Y.K. Agarwal.  $k$ -partition facets of the network design problem. *Networks*, 47(3):123–139, 2006.
- [2] R. Ahuja and J. Orlin. Inverse optimization. *Operations Research*, 49:771–783, 2001.
- [3] R. Ahuja and J. Orlin. Combinatorial algorithms for inverse network flow problems. *Networks*, 40:181–187, 2002.
- [4] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.
- [5] D. Alevras, M. Grötschel, and R. Wessäly. A network dimensioning tool. ZIB Technical Report SC-96-49, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1996.
- [6] D. Alevras, M. Grötschel, and R. Wessäly. Cost efficient network synthesis from leased lines. *Annals of Operations Research*, 76:1–20, 1998.
- [7] M. Andrews. Hardness of buy-at-bulk network design. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, 2004.
- [8] A. Atamtürk. On capacitated network design cut-set polyhedra. *Mathematical Programming*, 92:425–437, 2002.
- [9] Discnet. *discnet – Network optimization software library*. atesio GmbH, Sophie-Taeuber-Arp-Weg 27, D-12205 Berlin, Germany, 2000–2005. URL <http://www.atesio.de>.
- [10] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties*. Springer-Verlag, 1999.
- [11] B. Awerbuch and Y. Azar. Buy-at-bulk network design. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, Miami Beach FL*, pages 542–547, 1997.

- [12] J. Azevedo, J. Madeira, E. Martins, and F. Pires. A shortest paths ranking algorithm. In *Proceedings of Annual Conference Associazione Italiana di Ricerca Operativa (AIRO 1990), Sorrento, Italy*, pages 1001–1011, October 1990.
- [13] E. Balas. Facets of the knapsack polytope. *Mathematical Programming*, 8:146–164, 1975.
- [14] M. Ball, T. Magnanti, C. Monma, and G. Nemhauser, editors. *Handbooks in Operations Research and Management Science, 8: Network Routing*. North-Holland, Amsterdam, 1995.
- [15] F. Barahona. Network design using cut inequalities. *SIAM Journal on Optimization*, 6:823–837, 1996.
- [16] C. Barnhart, N.L. Boland, L.W. Clarke, E.L. Johnson, G.L. Nemhauser, and R.G. Sheno. Flight string models for aircraft fleetings and routing. *Transportation Science*, 32:208–220, 1998.
- [17] C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, and P. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1998.
- [18] C. Barnhart, C. Hane, and P. Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, 2000.
- [19] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas TX*, 1998.
- [20] M. Belaidouni and W. Ben-Ameur. A superadditive approach to solve the minimum cost single path routing problem: Preliminary results. In *Proceedings of the First International Network Optimization Conference (INOC 2003), Paris, France*, pages 67–71, Evry/Paris, October 2003.
- [21] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.
- [22] R. Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958.
- [23] R. Bellman and S. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1962.
- [24] R. Bellman and R. Kalaba. *Dynamic Programming and Modern Control Theory*. Academic Press, New York, London, 1965.



- [25] W. Ben-Ameur. Constrained length connectivity and survivable networks. *Networks*, 36:17–33, 2000.
- [26] W. Ben-Ameur. Multi-hour design of survivable classical IP networks. *International Journal of Communication Systems*, 15:553–572, 2002.
- [27] W. Ben-Ameur and E. Gourdin. Internet routing and related topology issues. *SIAM Journal on Discrete Mathematics*, 17(1):18–49, 2003.
- [28] W. Ben-Ameur, E. Gourdin, B. Liao, and N. Michel. Optimizing administrative weights for efficient single-path routing. In *Proceedings of Networks 2000*, 2000.
- [29] C. Berge. *Graphs and Hypergraphs*. North-Holland, Amsterdam, 1973.
- [30] D. Bienstock and O. Günlük. Capacitated network design – polyhedral structure and computation. *INFORMS Journal on Computing*, 8(3): 243–259, 1996.
- [31] D. Bienstock, S. Chopra, O. Günlük, and C.Y. Tsai. Minimum cost capacity installation for multicommodity flows. *Mathematical Programming*, 81:177–199, 1998.
- [32] R. Bixby and W. Cunningham. *Handbooks of Combinatorics*, pages 551–609. Elsevier, 1995. Chapter on “Matroids Optimization and Algorithms”.
- [33] A. Bley. Network optimization for IP-networks. Talk at the International Symposium on Mathematical Programming 2000, Atlanta, 2000.
- [34] A. Bley. A Lagrangian approach for integrated network design and routing in IP networks. In *Proceedings of the First International Network Optimization Conference (INOC 2003), Paris, France*, pages 107–113, October 2003.
- [35] A. Bley. Inapproximability results for the inverse shortest paths problem with integer lengths and unique shortest paths. ZIB Preprint ZR-05-04, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2005.
- [36] A. Bley and T. Koch. Integer programming approaches to access and backbone IP-network planning. ZIB Preprint ZR-02-41, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2002.
- [37] A. Bley, M. Grötschel, and R. Wessäly. Design of broadband virtual private networks: Model and heuristics for the B-WiN. In N. Dean, D.F. Hsu, and R. Ravi, editors, *Robust Communication Networks: Interconnection and Survivability*, volume 53 of *DIMACS Series in*

- Discrete Mathematics and Theoretical Computer Science*, pages 1–16. American Mathematical Society, 1998.
- [38] I. Bomze, M. Budinich, P. Pardalos, and M. Pelillo. The maximum clique problem. In D. Du and P. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 4. Kluwer Academic Publishers, 1999.
- [39] J.A. Bondy and U.S.R. Murty. *Graph Theory with Applications*. American Elsevier, New York, and Macmillan, London, 1976.
- [40] R. Borndörfer and Z. Kormos. An algorithm for maximum cliques. Unpublished working paper, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1997.
- [41] N. Bourquie, W. Ben-Ameur, E. Gourdin, and P. Tolla. Optimal shortest path routing for Internet networks. In *Proceedings of the First International Network Optimization Conference (INOC 2003)*, Paris, France, pages 119–125, 2003.
- [42] E.A. Boyd. Polyhedral results for the precedence-constrained knapsack problem. *Discrete Applied Mathematics*, 41:185–200, 1993.
- [43] B. Brockmüller, O. Günlük, and L.A. Wolsey. Designing private line networks - polyhedral analysis and computation. Technical Report CORE 9647, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 1996.
- [44] P. Broström. *Optimization in the Design of OSPF Telecommunication Networks*. PhD thesis, Linköping University, 2004.
- [45] P. Broström and K. Holmberg. Determining the non-existence of a compatible OSPF metric. Technical Report LiTH-MAT-R-2004-06, Linköping University, April 2004.
- [46] P. Broström and K. Holmberg. Stronger necessary conditions for the existence of a compatible OSPF metric. Technical Report LiTH-MAT-R-2004-08, Linköping University, May 2004.
- [47] C.A. Burdet and E.L. Johnson. A superadditive approach to solve linear integer programs. *Annals of Discrete Mathematics*, 1:117–144, 1977.
- [48] L. Buriol, M. Resende, C. Ribeiro, and M. Thorup. A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing. *Networks*, 46:36–56, 2005.
- [49] D. Burton. *On the Inverse Shortest Path Problem*. PhD thesis, Department of Mathematics, Facultés Universitaires ND de la Paix, Namur, Belgium, 1993.

- [50] D. Burton and P. Toint. On an instance of the inverse shortest paths problem. *Mathematical Programming*, 53:45–61, 1992.
- [51] D. Burton and P. Toint. On the use of an inverse shortest paths algorithm for recovering linearly correlated costs. *Mathematical Programming*, 63:1–22, 1994.
- [52] D. Burton, B. Pulleyblank, and P. Toint. The inverse shortest path problem with upper bounds on shortest path costs. In P. Pardalos, D. Hearn, and W. Hager, editors, *Network Optimization*, volume 450 of *Lecture Notes in Economics and Mathematical Systems*, pages 156–171. Springer-Verlag, 1997.
- [53] M.-C. Cai and Y. Li. Inverse matroid intersection problem. *Mathematical Methods of Operations Research*, 45:235–243, 1997.
- [54] R. Callon. Use of OSI IS-IS for routing in TCP/IP and dual environments. IETF Internet RFC 1195, December 1990. URL <http://www.ietf.org/rfc/rfc1195.txt>.
- [55] R. Carr, L. Fleischer, V. Leung, and C. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms 2000*, pages 106–115, 2000.
- [56] M. Charikar and A. Karagiozova. On non-uniform multicommodity buy-at-bulk network design. In *Proceedings of the Thirty-seventh Annual ACM Symposium on the Theory of Computing, Baltimore, Maryland, USA*, 2005.
- [57] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S.A. Plotkin. Approximating a finite metric by a small number of tree metrics. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, Palo Alto CA*, pages 379–388, 1998.
- [58] S. Chopra and M. Rao. The Steiner tree problem I: Formulations, compositions and extension of facets. *Mathematical Programming*, 64(2):209–229, 1994.
- [59] S. Chopra and M. Rao. The Steiner tree problem II: Properties and classes of facets. *Mathematical Programming*, 64(2):231–246, 1994.
- [60] K. Coffman and A. Odlyzko. *Handbook of Massive Data Sets*, pages 47–93. Kluwer Academic Publishers, 2001. Chapter Internet Growth: Is there a “Moore’s Law” for data traffic?
- [61] M. Conforti and M. Laurant. On the facial structure of independence system polyhedra. *Mathematics of Operations Research*, 13(3), 1988.

- [62] G. Cornuéjols and A. Sassano. On the 0,1 facets of the set covering polytope. *Mathematical Programming*, 43:45–55, 1989.
- [63] S. Cosares and I. Saniee. An optimization problem related to balancing loads on SONET rings. *Telecommunication Systems*, 3:165–182, 1994.
- [64] CPLEX 10.1. CPLEX 10.1 reference manual, 2006. URL <http://www.cplex.com>.
- [65] P. Crescenzi and V. Kann, editors. *A Compendium of NP Optimization Problems*. 2003. URL <http://www.nada.kth.se/~viggo/problemlist/compendium.html>.
- [66] G. Dahl and M. Stoer. A polyhedral approach to multicommodity survivable network design. *Numerische Mathematik*, 68(1):149–167, 1994.
- [67] G. Dahl and M. Stoer. A cutting plane algorithm for multicommodity survivable network design problems. *INFORMS Journal on Computing*, 10(1):1–11, 1998.
- [68] G. Dahl, A. Martin, and M. Stoer. Routing through virtual paths in layered telecommunication networks. *Operations Research*, 47(5):693–702, 1999.
- [69] L. de Giovanni, B. Fortz, and M. Labbé. A lower bound for the Internet protocol network design problem. In *Proceedings of the Second International Network Optimization Conference (INOC 2005), Lisbon, Portugal*, volume 1, pages 402–408, March 2005.
- [70] J. Desrosiers and M.E. Lübbecke. A primer in column generation. Technical Report 2003/48, Technische Universität Berlin, 2003.
- [71] J. Desrosiers and M.E. Lübbecke. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.
- [72] *Deutsche Forschungsnetz Verein*. DFN. URL <http://www.dfn.de>.
- [73] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [74] Y. Dinitz, N. Garg, and M. Goemans. On the single source unsplittable flow problem. *Combinatorica*, 19:1–25, 1999.
- [75] Y. Dodis and S. Khanna. Designing networks with bounded pairwise distance. In *Proceedings of the Thirty-first Annual ACM Symposium on the Theory of Computing, Atlanta GA*, pages 750–759, 1999.

- [76] J. Edmonds. *Combinatorial Structures and their Applications*, pages 69–87. Gordon and Breach, New York, 1970. Chapter on Submodular Functions, Matroids, and Certain Polyhedra.
- [77] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1:127–223, 1971.
- [78] D. Eppstein. Finding the  $k$  shortest paths. Technical Report TR-94-26, Department of Information and Computer Science, University of California, May 1994.
- [79] P. Erdős and H. Sachs. Regüäre Graphen gegebener Taillenweite mit minimaler Knotenzahl. *Wissenschaftliche Zeitung der Universität Halle-Wittenberg*, 12:251–257, 1963.
- [80] A. Eremin, F. Ajili, and R. Rodosek. A set-based approach to the optimal IGP weight setting problem. In *Proceedings of the Second International Network Optimization Conference (INOC 2005), Lisbon, Portugal*, volume 1, pages 386–392, March 2005.
- [81] M. Ericsson, M. Resende, and P. Pardalos. A genetic algorithm for the weight setting problem in OSPF routing. *Journal of Combinatorial Optimization*, 6:299–333, 2002.
- [82] R. Euler, M. Jünger, and G. Reinelt. Generalizations of cliques, odd cycles and anticycles and their relation to independence systems polyhedra. *Mathematics of Operations Research*, 12(3):451–462, 1987.
- [83] L. Fadejeva. Ein Column-Generation-Ansatz zur Kostenoptimierung von ausfallsicheren Kommunikationsnetzen mit Single-Path-Routing. Diploma thesis, Technische Universität Berlin, August 2003.
- [84] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the Thirty-fifth Annual ACM Symposium on the Theory of Computing*, 2003.
- [85] A. Farago, A. Szentesi, and B. Szviatovszki. Allocation of administrative weights in PNNI. In *Proceedings of Networks '98, Sorrento, Italy*, pages 621–625, 1998.
- [86] A. Farago, A. Szentesi, and B. Szviatovszki. Inverse optimization in high-speed networks. *Discrete Applied Mathematics*, 129:83–98, 2003.
- [87] G. Farkas. Über die Anwendung des mechanischen Principes von Fourier. *Mathematische und naturwissenschaftliche Berichte aus Ungarn*, 12:263–281, 1895.

- 
- [88] G. Farkas. Theorie der einfachen Ungleichungen. *Journal für reine und angewandte Mathematik*, 124:1–27, 1902.
- [89] S. Fekete, W. Hochstättler, S. Kromberg, and C. Moll. The complexity of an inverse shortest path problem. In R. Graham, J. Kratochvíl, J. Nešetřil, and F. Roberts, editors, *Contemporary Trends in Discrete Mathematics: From DIMACS and DIMATIA to the Future*, volume 49, pages 113–127. American Mathematical Society, 1999.
- [90] O.E. Flippo, A.W.J. Kolen, A.M.C.A. Koster, and R.L.M.J. van de Leensel. A dynamic programming algorithm for the local access telecommunication network expansion problem. *European Journal of Operational Research*, 127:189–202, 2000.
- [91] L.R. Ford and D.R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, New Jersey, 1962.
- [92] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In *Proceedings of the 19th IEEE Infocom 2000, Tel-Aviv, Israel*, pages 519–528, 2000.
- [93] B. Fortz and M. Thorup. Increasing Internet capacity using local search. *Computational Optimization and Applications*, 29(1):13–48, 2004.
- [94] A. Frank. Packing paths, circuits, and cuts – A survey. In B. Korte, L. Lovász, H.J. Prömel, and A. Schrijver, editors, *Algorithms and Combinatorics*, volume 9, chapter 4, pages 47–100. Springer-Verlag, 1990.
- [95] A. Frank. *Handbooks of Combinatorics*, pages 111–177. Elsevier, 1995. Chapter Connectivity and Network Flows.
- [96] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman and Company, New York, 1979.
- [97] F. Gavril. Algorithms on circular arc graphs. *Networks*, 4:357–369, 1974.
- [98] M. Goemans. The Steiner tree polytope and related polyhedra. *Mathematical Programming*, 63(2):157–182, 1994.
- [99] M.X. Goemans and D.P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24:296–317, 1995.
- [100] T. Griffin and G. Wilfong. An analysis of BGP convergence properties. In *SIGCOM*, 1999.

- [101] M. Grötschel and L. Lovász. *Handbooks of Combinatorics*, chapter Combinatorial Optimization, pages 1541–1597. Elsevier, 1995.
- [102] M. Grötschel and C. Monma. Integer polyhedra associated with certain network design problems with connectivity constraints. *SIAM Journal on Discrete Mathematics*, 3:502–523, 1990.
- [103] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1988.
- [104] M. Grötschel, C. Monma, and M. Stoer. Facets for polyhedra arising in the design of communication networks with low-connectivity constraints. *SIAM Journal on Optimization*, 2(3):474–504, August 1992.
- [105] M. Grötschel, C. Monma, and M. Stoer. *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, chapter 10, pages 617–672. Elsevier, North-Holland, Amsterdam, 1995. M. Ball, T. Magnanti, C. Monma, G. Nemhauser (Eds.).
- [106] M. Grötschel, A. Martin, and R. Weismantel. Packing Steiner trees: Polyhedral investigations. *Mathematical Programming*, 72:101–123, 1996.
- [107] M. Grötschel, A. Martin, and R. Weismantel. Packing Steiner trees: Further facets. *European Journal on Combinatorics*, 17:39–52, 1996.
- [108] S. Guha, A. Meyerson, and K. Munagala. A constant factor approximation for the single sink edge installation problem. In *Proceedings of the Thirty-third Annual ACM Symposium on the Theory of Computing, Hersonissos, Crete, Greece*, pages 383–388, 2001.
- [109] O. Günlük. A branch-and-cut algorithm for capacitated network design problems. *Mathematical Programming*, 86:17–39, 1999.
- [110] A. Gupta, A. Kumar, and T. Roughgarden. Simpler and better approximation algorithms for network design. In *Proceedings of the Thirty-fifth Annual ACM Symposium on the Theory of Computing*, pages 365–372, 2003.
- [111] P. Hammer, E. Johnson, and U. Peled. Facets of regular 0-1 polytopes. *Mathematical Programming*, 8:179–206, 1975.
- [112] J. Håstad. Some optimal inapproximability results. In *Proceedings of the Twenty-ninth Annual ACM Symposium on the Theory of Computing, El Paso TX*, pages 1–10, 1997.
- [113] D. Hausmann, T. Jenkyns, and B. Korte. Worst case analysis of greedy type algorithms for independence systems. Technical report, Institut für Ökonometrie und Operations Research, Universität Bonn, 1977.

- 
- [114] C. Hedrick. Routing information protocol. IETF Internet RFC 1058, June 1988. URL <http://www.ietf.org/rfc/rfc1058.txt>.
  - [115] K. Holmberg and D. Yuan. Optimization of Internet protocol network design and routing. *Networks*, 43(1):39–53, 2004.
  - [116] T. Hu. Multi-commodity network flows. *Operations Research*, 11: 344–360, 1963.
  - [117] M. Iri. On an extension of the maximum-flow minimum-cut theorem to multicommodity flows. *Journal of the Operations Research Society of Japan*, 13(3):129–135, 1971.
  - [118] T. Jenkyns. The efficacy of the greedy algorithm. In *Proceedings of the Seventh Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pages 341–350, 1976.
  - [119] R.G. Jeroslow. Cutting plane theory: Algebraic methods. *Discrete Mathematics*, 23:121–150, 1978.
  - [120] D.S. Johnson and K.A. Niemi. On knapsacks, partitions, and a new dynamic programming technique for trees. *Mathematics of Operations Research*, 1983.
  - [121] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.
  - [122] N. Katoh, T. Ibaraki, and H. Mine. An efficient algorithm for  $k$  shortest simple paths. *Networks*, 12:411–427, 1982.
  - [123] J.M. Kleinberg. Single-source unsplittable flow. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science, Burlington VT*, pages 68–77, 1996.
  - [124] S. Kolliopoulos and C. Stein. Improved approximation algorithms for unsplittable flow problems. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, Miami Beach FL*, pages 426–435, 1997.
  - [125] B. Korte and D. Hausmann. An analysis of the greedy heuristic for independence systems. *Annals of Discrete Mathematics*, 2:65–74, 1978.
  - [126] B. Korte, L. Lovász, and R. Schrader. *Greedoids*. Springer-Verlag, 1991.
  - [127] A. Kröller. Network optimization: Integration of hardware configuration and capacity dimensioning. Diploma thesis, Technische Universität Berlin, June 2003.



- [128] M. Labbé and H. Yaman. Polyhedral analysis for concentrator location problems. Technical Report 2003/13, Université Libre de Bruxelles, 2003.
- [129] M. Labbé and H. Yaman. Solving the uncapacitated concentrator location problem with star routing. Technical Report 2003/15, Université Libre de Bruxelles, 2003.
- [130] M. Labbé and H. Yaman. Projecting the flow variables for hub location problems. *Networks*, 44:84–93, 2004.
- [131] M. Laurent. A generalization of antiwebs to independence systems and their canonical facets. *Mathematical Programming*, 45:97–108, 1989.
- [132] Leda. LEDA: Library of Efficient Data types and Algorithms, 1998–2003. URL <http://www.algorithmic-solutions.com>.
- [133] F. Lin and J. Wang. Minimax open shortest path first routing algorithms in networks supporting the SMDS service. In *Proceedings of the IEEE International Conference on Communications 1993 (ICC'93)*, Geneva, Suisse, volume 2, pages 666–670, 1993.
- [134] D. Lorenz, A. Orda, D. Raz, and Y. Shavitt. How good can IP routing be? Technical Report 2001-17, DIMACS, Center for Discrete Mathematics and Theoretical Computer Science, Rutgers University, Princeton University, AT&T Bell Laboratories and Bellcore, 2001.
- [135] T. Magnanti and S. Raghavan. Strong formulations for network design problems with connectivity requirements. unpublished manuscript, 1999.
- [136] T. Magnanti, P. Mirchandani, and R. Vachani. Modeling and solving the two-facility capacitated network loading problem. *Operations Research*, 43(1):142–157, 1995.
- [137] Y. Mansur and D. Peleg. An approximation algorithm for minimum-cost network design. Technical Report CS94-22, Weizman Institute of Science, Rehovot, Israel, 1994.
- [138] H. Marchand and L.A. Wolsey. The 0 – 1 knapsack problem with a single continuous variable. *Mathematical Programming*, 85(1):15–34, May 1999.
- [139] E. Martins and J. Santos. A new shortest paths ranking algorithm. *Investigação Operacional*, 20(1):47–62, 2000.
- [140] A. Meyerson and K. Munagala S. Plotkin. Cost-distance: Two-metric network design. In *Proceedings of the Thirty-second Annual ACM*

- Symposium on the Theory of Computing, Portland OR*, pages 624–630, 2000.
- [141] J. Milbrandt. Possibilities of routing optimization in IP networks. Master's thesis, Department of Computer Science, University of Würzburg, 2001.
  - [142] C. Moll. *Das inverse Kürzeste-Wege-Problem*. PhD thesis, Universität zu Köln, 1995.
  - [143] J. Moy. OSPF version 2. IETF Internet RFC 2328, April 1998. URL <http://www.ietf.org/rfc/rfc2328.txt>.
  - [144] R. Müller and A. Schulz. Transitive packing: A unifying concept in combinatorial optimization, 1999.
  - [145] G. Nemhauser and L. Trotter. Properties of vertex packing and independence system polyhedra. *Mathematical Programming*, 6, 1974.
  - [146] G. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.
  - [147] K. Onaga and O. Kakusho. On feasibility conditions of multicommodity flows in networks. *Transactions on circuit theory*, 18(4):425–429, 1971.
  - [148] S. Orłowski. Local and global restoration of node and link failures in telecommunication networks. Master's thesis, Technische Universität Berlin, February 2003.
  - [149] S. Orłowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0—Survivable Network Design Library. In *Proceedings of the Third International Network Optimization Conference (INOC 2007), Spa, Belgium*, April 2007. <http://sndlib.zib.de>.
  - [150] P. Orponen and H. Mannila. On approximation preserving reductions: Complete problems and robust measures. Technical Report C-1987-28, Department of Computer Science, University of Helsinki, 1987.
  - [151] M.W. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5:199–215, 1973.
  - [152] M.W. Padberg. A note on zero-one programming. *Operations Research*, 23:883–837, 1975.
  - [153] M.W. Padberg. (1,k)-configurations and facets for packing polyhedra. *Mathematical Programming*, 18:94–99, 1980.

- [154] C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- [155] C.H. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *Proceedings of the Twentieth Annual ACM Symposium on the Theory of Computing, Chicago IL*, pages 229–234, 1988.
- [156] C.H. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *Journal on Computer System Science*, 43:425–440, 1991.
- [157] K. Park and S. Park. Lifting cover inequalities for the precedence-constrained knapsack problem. *Discrete Applied Mathematics*, 72:219–241, 1997.
- [158] A. Parmar, S. Ahmed, and J. Sokol. An integer programming approach to the OSPF weight setting problem. *Optimization Online*, 2005.
- [159] R. Perlman. *Interconnections: Bridges, Routers, Switches and Inter-networking Protocols*. Addison-Wesley, 2nd edition, 1999.
- [160] T. Pfender. Arboreszenz-Flüsse in Graphen: polyedrische Untersuchungen. Master’s thesis, Technische Universität Berlin, 2000.
- [161] M. Pfetsch. *The Maximum Feasible Subsystem Problem and Vertex-Facet Incidences of Polyhedra*. PhD thesis, Technische Universität Berlin, 2002.
- [162] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers, 2004.
- [163] M. Pióro, A. Szentesi, J. Harmatos, and A. Jüttner. On OSPF related network optimization problems. In *8th IFIP Workshop on Performance Modelling and Evaluation of ATM & IP Networks*, pages 70/1–70/14, Ilkley, UK, 2000.
- [164] PNNI. Private Network-Network Interface specification version 1.0 (PNNI). The ATM Forum Technical Committee, af-pnni-0055.000., March 1996.
- [165] J. Postel. User datagram protocol. IETF Internet RFC 768, August 1980. URL <http://www.ietf.org/rfc/rfc0768.txt>.
- [166] J. Postel. Internet protocol. IETF Internet RFC 791, September 1981. URL <http://www.ietf.org/rfc/rfc791.txt>.

- 
- [167] J. Postel. Internet control message protocol. IETF Internet RFC 792, September 1981. URL <http://www.ietf.org/rfc/rfc792.txt>.
  - [168] J. Postel. Transmission control protocol. IETF Internet RFC 793, September 1981. URL <http://www.ietf.org/rfc/rfc793.txt>.
  - [169] M. Prytz. *On Optimization in Design of Telecommunications Networks with Multicast and Unicast Traffic*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2002.
  - [170] Y. Rekhter and P. Gross. Application of the border gateway protocol in the Internet. IETF Internet RFC 1772, March 1995. URL <http://www.ietf.org/rfc/rfc1772.txt>.
  - [171] Y. Rekhter and T. Li. Border gateway protocol 4 (BGP-4). IETF Internet RFC 1771, March 1995. URL <http://www.ietf.org/rfc/rfc1771.txt>.
  - [172] M.G.C. Resende and P.M. Pardalos, editors. *Handbook of Optimization in Telecommunications*. Springer-Verlag, 2006.
  - [173] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. IETF Internet RFC 3031, January 2001. URL <http://www.ietf.org/rfc/rfc3031.txt>.
  - [174] A. Sassano. On the facial structure of the set covering polytope. *Mathematical Programming*, 44:181–202, 1989.
  - [175] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, 1986.
  - [176] A. Schrijver. Homotopic routing methods. In B. Korte, L. Lovász, H.J. Prömel, and A. Schrijver, editors, *Algorithms and Combinatorics*, volume 9, chapter 12, pages 329–371. Springer-Verlag, 1990.
  - [177] A. Schrijver, P. Seymour, and P. Winkler. The ring loading problem. *SIAM Journal on Applied Mathematics*, 11(1):1–14, February 1998.
  - [178] C. Skiscim and B. Golden. Solving k-shortest and constrained shortest path problems efficiently. *Annals of Operations Research*, 20:249–282, 1989.
  - [179] M. Skutella. Approximating the single source unsplittable min-cost flow problem. *Mathematical Programming*, 91(3):493–514, 2002.
  - [180] D. Staehle, S. Köhler, and U. Kohlhaas. Towards an optimization of the routing parameters for ip networks. Technical Report TR 258, Department of Computer Science, University of Würzburg, 2000.

- [181] W. Stevens. *TCP/IP Illustrated*, volume 1. Addison-Wesley, 1994.
- [182] K. Talwar. Single-sink buy-at-bulk LP has constant integrality gap. In *Proceedings of the 9th Conference on Integer Programming and Combinatorial Optimization (IPCO 2002)*, pages 475–486, 2002.
- [183] R. Tarjan. Sensitivity analysis of minimum spanning trees and shortest path trees. *Information Processing Letters*, 14(1):30–33, 1982.
- [184] A. Tomaszewski, M. Pióro, M. Dzida, and M. Zagożdżon. Optimization of administrative weights in IP networks using the branch-and-cut approach. In *Proceedings of the Second International Network Optimization Conference (INOC 2005), Lisbon, Portugal*, volume 2, pages 393–400, March 2005.
- [185] C. Tong and K. Lam. An embedded connectionist approach for the inverse shortest paths problem. Technical report, Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, 1996.
- [186] R. van de Leensel. *Models and Algorithms for Telecommunication Network Design*. PhD thesis, Faculty of Economics and Business Administration, Maastricht University, The Netherlands, 1999.
- [187] R.L.M.J. van de Leensel, C.P.M. van Hoesel, and J.J. van de Klundert. Lifting valid inequalities for the precedence constrained knapsack problem. *Mathematical Programming*, 86:161–186, 1999.
- [188] C. van Hoesel, A.M.C.A. Koster, R. van de Leensel, and M. Savelsbergh. Polyhedral results for the edge capacity polytope. *Mathematical Programming*, 92(2):335–358, 2002.
- [189] D. Welsh. *Handbooks of Combinatorics*, chapter Matroids: Fundamental Concepts, pages 481–526. Elsevier, 1995.
- [190] R. Wessäly. *Dimensioning Survivable Capacitated NETWORKS*. PhD thesis, Technische Universität Berlin, April 2000.
- [191] G. Wilfong and P. Winkler. Ring routing and wavelength translation. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms 1998*, pages 333–341, 1998.
- [192] L.A. Wolsey. Faces of linear inequalities in 0-1 variables. *Mathematical Programming*, 8:165–178, 1975.
- [193] L.A. Wolsey. Valid inequalities for 0/1 knapsacks and MIPs with Generalized Upper Bound constraints. *Discrete Applied Mathematics*, 29:251–261, 1990.

- [194] L.A. Wolsey. *Integer Programming*. John Wiley & Sons, 1998.
- [195] S. Xu and J. Zhang. An inverse problem of the weighted shortest path problem. *Japan Journal of Industrial Applied Mathematics*, 12:47–59, 1995.
- [196] J. Yen. Finding the  $k$  shortest loopless paths in a network. *Management Science*, 17:712–716, 1971.
- [197] M. Zagożdżon, M. Dzida, M. Pióro, and A. Tomaszewski. Optimization of administrative weights in single path networks with OSPF routing. In *12th Polish Teletraffic Symposium PSRT 2005, Poznań*, September 2005.
- [198] J. Zhang and Z. Liu. Calculating some inverse linear programming problems. *Journal of Computational and Applied Mathematics*, 72: 261–273, 1996.

# Index

- 2-band inequality, 203
- admissible metric, 230
- admissible path, 230
- admissible unsplittable shortest path routing, 230
- anti-arborescence, 12
- approximation algorithm, 17
- approximation preserving reduction, 19
- approximation ratio, 17
- $\mathcal{APX}$ , 18
- arborescence, 12
- arc routing, 167
- AS, *see* autonomous system
- associated forwarding, *see* forwarding
- autonomous system, 22
- B-property, *see* Bellman property
- band inequality, 203
- Bellman property, 75
- BGP, *see* Border Gateway Protocol
- Border Gateway Protocol, 23
- BUY-AT-BULK NETWORK DESIGN problem, 142
- CAP-USPR problem (capacitated network design with unsplittable shortest path routing), 35
- capacitated network design problem with unsplittable shortest path routing, 35
- circuit, 13
- circuit hypergraph, 14
- circuit inequality
  - for arc routing variables, 168
  - for path variables, 164
- clique inequality
  - for arc routing variables, 199
  - for path variables, 197
- commodities, 32, 229
- compatible metric, *see* metric
- conflict hypergraph, 73
- conflicting paths, 75, *see also* Bellman property
- congestion, 30, 34, 232
- cut inequality, 202
- cyclic comparability, 98
- delay, 30
- demands, 32, 229
- dimensioning, 28, 226
- directed cut inequality, 202
- distance vector protocol, 24
- dominance among metrics, 142
- ECMP, *see* equal cost multi path routing
- equal cost multi path routing, 25, *see* shortest multi-path routing
- EXACT SET COVER problem, 47
- facet, 9
- Farkas Lemma, 9
- Farkas ray, 80
- FC-USPR problem (fixed charge network design with unsplittable shortest path routing), 36
- feasible potential, 62
- fixed charge network design problem with unsplittable shortest path routing, 36
- forwarding, 60
  - associated with arc routing, 168
  - associated with path set, 61
- forwarding arc, 60
- forwarding table, 26
- FULLY DISJOINT PATHS problem, 124
- GB-property, *see* generalized Bellman property

- generalized Bellman property, 77
- generalized cyclic comparability, 78,
  - see also* generalized Bellman property
- GENERALIZED STEINER NETWORK problem, 129
- graph partition inequality, 202
- hat-cycle graph, 100
- hop limits, 229
- IIS, *see* irreducibly inconsistent system
- in-degree inequality, 200
- independence system, 13
- independence system polytope, 194, 198
- induced  $(1, k)$ -configuration inequality, 215
- induced  $k$ -cover inequality, 214
- induced cover inequality, 213
- induced flow, 34
- inter-domain routing, 23
- Intermediate System to Intermediate System protocol, 24
- Internet Protocol, 22
- intra-domain routing, 24
- inverse shortest paths problem, 40, *see also* inverse unique shortest paths problem
- inverse unique shortest path forwarding problem
  - feasibility, 62
  - minimum arc length, 66
  - minimum total length, 66
- inverse unique shortest paths problem
  - feasibility, 43
  - minimum arc length, 43
  - minimum path length, 43
  - minimum total length, 66
- IP, *see* Internet Protocol
- IP routing, *see* shortest path routing
- irreducibly inconsistent system, 80
- IS-IS, *see* Intermediate System to Intermediate System protocol
- IUSP problem (inverse unique shortest paths), 43
- IUSPF problem (inverse unique shortest path forwarding), 62
- jitter, 30
- knapsack polytope, 210
- link graph, 222
- link state protocol, 24
- load group, 232
- loss-rate, 30
- MAX-3-SAT problem, 94
- MAX-USPS problem (maximum unique shortest path system), 91
- MAX-WEIGHT-USPS problem (maximum weight unique shortest path system), 91
- MAXIMUM WEIGHT CLIQUE problem, 102
- metric, 33
  - admissible, 230
  - compatible
    - with forwarding, 61
    - with path set, 42
  - dominance, 142
  - stretch, 142
  - tree  $\sim$ , 142
  - valid, 33
- metric inequality, 202
- MIN-ARC-IUSP problem (minimum arc length inverse unique shortest paths), 43
- MIN-ARC-IUSPF problem (minimum arc length inverse unique shortest path forwarding), 66
- MIN-CON-USPR problem (minimum congestion unsplittable shortest path routing), 34
- MIN-NON-USPF problem (minimum non-unique shortest path forwarding), 104
- MIN-NON-USPS problem (minimum non-unique shortest path system), 81
- MIN-PATH-IUSP problem (minimum path length inverse unique shortest paths), 43
- MIN-SUM-IUSP problem (minimum total length inverse unique shortest paths), 66
- MIN-SUM-IUSPF problem (minimum total length inverse unique shortest path forwarding), 66



- MIN-WEIGHT-NON-USPF problem (minimum weight non-unique shortest path forwarding), 105
- MIN-WEIGHT-NON-USPS problem (minimum weight non-unique shortest path system), 81
- MIN-WEIGHT-SAT problem, 133
- minimum congestion unsplittable shortest path routing problem, 34
- MINIMUM VERTEX COVER problem, 84
- multicommodity flow routing, 120
- ND-USPR problem (network design with unsplittable shortest path routing), 233
- network design problem, *see* CAP-USPR, FC-USPR and ND-USPR
- no-bottleneck condition, 123
- non-decreasing function, 204
- $\mathcal{NP}$ , 16
- $\mathcal{NPO}$ , 16
- Open Shortest Path First protocol, 24
- OSPF, *see* Open Shortest Path First protocol
- out-degree inequality, 199
- $\mathcal{P}$ , 16
- PARTITION problem, 119
- partition band inequality, 203
- performance ratio, 17
- $\mathcal{PO}$ , 17
- POINT-TO-POINT CONNECTION problem, 129
- precedence constrained knapsack polytope, 211
- precedence constrained knapsack problem, 217
- precedence constraint, 211
- PRICE-USPR-PATH problem (pricing of path variables in path routing models), 178
- rank
  - of a forwarding, 103
  - of a path set, 73
- rank inequality
  - for arc routing variables, 199
  - for path variables, 196
- rank quotient, 92
- RING LOADING problem, 144
- RIP, *see* Routing Information Protocol
- routing, 28, 32
- routing domain, *see* autonomous system
- Routing Information Protocol, 24
- routing lengths, *see* metric
- routing metric, *see* metric
- routing weights, *see* metric
- SEP-IUSP problem (separation of shortest path uniqueness constraints), 45
- SEP-USPF-CIRCUIT problem (separation of circuit inequalities for arc routing variables), 184
- SEP-USPS-CIRCUIT problem (separation of circuit inequalities for path routing variables), 176
- SET PARTITION problem, 47
- shortest multi-path routing, 25, 120
- shortest path routing, *see also* unsplittable shortest path routing, 24–276
- source-invariant routing, 120
- stable set, 11
- stable set polytope, 195, 198
- strengthened 2-band inequality, 203
- strengthened cut inequality, 203
- strengthened metric inequality, 203
- stretch of a metric, 142
- sub-optimality of paths, *see* Bellman property
- subpath inequality, 198, 200, 201
- subpath property, *see* Bellman property
- superadditive function, 204
- superadditive metric inequality, 204, 207
- supply graph, 32, 222
- taboo nodes, 229
- TE-USPR problem (traffic engineering with unsplittable shortest path routing), 234
- tight arc, 62
- traffic engineering, 30

- traffic engineering problem, *see* MIN-CON-USPR and TE-USPR
- traffic splitting, *see* shortest multi-path routing
- tree metric, 142
- unique shortest path forwarding, 61
  - compatible metric, 61
  - inverse  $\sim$  problem, 62
  - irreducible non- $\sim$ , 103
  - maximal  $\sim$ , 103
  - minimum non- $\sim$  problem, 104
  - minimum weight non- $\sim$  problem, 105
- unique shortest path forwarding polytope, 172
- unique shortest path system, 42
  - compatible metric, 42
  - inverse unique shortest paths problem, 43
  - irreducible non- $\sim$ , 72
  - maximal  $\sim$ , 72
  - maximum  $\sim$  problem, 91
  - maximum weight  $\sim$  problem, 91
  - minimum non- $\sim$  problem, 81
  - minimum weight non- $\sim$  problem, 81
- unsplittable flow routing, 120
- unsplittable shortest path arc routing polytope, 167
- unsplittable shortest path routing, 33, *see also* unique shortest path system, unique shortest path forwarding
  - admissible, 230
- unsplittable shortest path routing polytope, 164
- unsplittable source-invariant routing, 120
- USPF, *see* unique shortest path forwarding
- USPR, *see* unsplittable shortest path routing
- USPS, *see* unique shortest path system
- valid metric, 33
- valid-cycle property, 81
- weakly disjoint paths, 57
- weakly stable set, 11