Université Libre de Bruxelles
Faculté des Sciences

# Design of Survivable Networks with Bounded-Length Paths

**David Huygens**

Thèse présentée en vue de l'obtention du grade de Docteur en Sciences, spécialité Informatique (Recherche Opérationnelle)

Juin 2005 Directeur de thèse : M. Labbé

*In memory of Nani,*
*who was my first teacher*

# Acknowledgments

This Ph.D. thesis has been realized at the *Université Libre de Bruxelles* with a scholarship from the *Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture.* I would like to thank these two institutions for their support.

First, I express my sincere gratitude to Professor Martine Labbé who led me to discover and love operations research. She taught me how to conduct my research and trusted me in the realization of this doctorate. Thanks to her, I had the opportunity to work on an interesting project for four years and to meet fellow researchers.

I would like to deeply thank Professor A. Ridha Mahjoub who welcomed me in Clermont-Ferrand so many times and was always able to find time to work with me. Without him, I am not sure that this thesis would be as it is today. He is not only a great working partner, but also an exceptional person on whom one can always count.

I thank Pierre Pesneau with whom I had the chance to work, and who has become a very good friend. My thoughts also go to Pierre Fouilhoux, Sylvie Borne, Lise Slama, and all other present and past members of the EPOC team that I met at the LIMOS (*Université Clermont-Ferrand II*).

Of course, most of my Ph.D. time was spent at the *Université Libre de Bruxelles.* Therefore, I would like to thank all the members of (and affiliates to) the Computer Science Department, and, in particular, the GOM and SMG. There are too many of them to name them all here (and I do not want to forget anyone), but I am sure that they will recognize themselves. It was great to work in such a good atmosphere, with interesting and funny people. I will miss our daily passionnate arguments.

Finally, I am very grateful to all my relatives for their constant support. In particular, I want to thank my wife, Romina, for our day-to-day life together. While doing her own Ph.D., she was still able to find the time to hearten me. I address all my love to my family, especially to my parents, who have given me, during my whole life, all I needed to succeed in this project. I thank all my friends for the great moments spent together, namely Grégory, Ana and Denis, but also the ones who, even far away, stay present in my heart.

# Abstract

In this thesis, we consider the *k-edge connected L-hop-constrained network design problem*. Given a weighted graph $G = (N, E)$, a set $D \subseteq N \times N$ of pairs of terminal nodes, and two integers $k, L \geq 2$, it consists in finding in $G$ the minimum cost subgraph containing at least $k$ edge-disjoint paths of at most $L$ edges between each pair in $D$. This problem is of great interest in today's telecommunication industry, where highly survivable networks need to be constructed.

We first study the particular case where the set of demands $D$ is reduced to a single pair $\{s, t\}$. We propose an integer programming formulation for the problem, which consists in the $st$-cut and trivial inequalities, along with the so-called $L$-$st$-path-cut inequalities. We show that these three classes of inequalities completely describe the associated polytope when $k = 2$ and $L = 2$ or 3, and give necessary and sufficient conditions for them to be facet-defining. We also consider the dominant of the associated polytope, and discuss how the previous inequalities can be separated in polynomial time.

We then extend the complete and minimal description obtained above to any number $k$ of required edge-disjoint $L$-$st$-paths, but when $L = 2$ only. We devise a cutting plane algorithm to solve the problem, using the previous polynomial separations, and present some computational results.

After that, we consider the case where there is more than one demand in $D$. We first show that the problem is strongly $NP$-hard, for all $L$ fixed, even when all the demands in $D$ have one root node in common. For $k = 2$ and $L = 2, 3$, we give an integer programming formulation, based on the previous constraints written for all pairs $\{s, t\} \in D$. We then proceed by giving several new classes of facet-defining inequalities, valid for the problem in general, but more adapted to the rooted case. We propose separation procedures for these inequalities, which are embedded within a Branch-and-Cut algorithm to solve the problem when $L = 2, 3$. Extensive computational results from it are given and analyzed for both random and real instances.

Since those results appear less satisfactory in the case of arbitrary demands (non necessarily rooted), we present additional families of valid inequalites in that situation. Again, separation procedures are devised for them, and added to our previous Branch-and-Cut algorithm, in order to see the practical improvement granted by them.

Finally, we study the problem for greater values of $L$. In particular, when $L = 4$, we propose new families of constraints for the problem of finding a subgraph that contains at least two $L$-$st$-paths either node-disjoint, or edge-disjoint. Using these, we obtain an integer programming formulation in the space of the design variables for each case.

**Key words :** Survivable network, edge connectivity, node connectivity, hop-constrained paths, integer programming, polytope, facet, Branch-and-Cut algorithm.

# Contents

**Introduction** **1**

**1 Preliminary Notions and Previous Works** **5**

1.1 Notions of graph theory . . . . . 5

1.2 Notions of polyhedral theory . . . . . 7

1.3 State of the art in network design . . . . . 9

**2 Two Edge-Disjoint Hop-Constrained Paths Problem** **17**

2.1 Introduction . . . . . 17

2.2 Formulation for $L = 2, 3$ . . . . . 20

2.3 THPP polytope for $L = 2, 3$ . . . . . 24

2.4 Structural properties . . . . . 29

2.5 Proof of Theorem 2.3.1 . . . . . 41

2.6 Facets of $P(G, L)$ . . . . . 44

2.7 Dominant of $P(G, L)$ . . . . . 51

2.8 Concluding remarks . . . . . 52

**3 $k$ Edge-Disjoint Hop-Constrained Paths Problem** **55**

3.1 Introduction . . . . . 55

3.2 Formulation for $L = 2, 3$ . . . . . 57

3.3 Facets and Completeness . . . . . 59

3.3.1 Facets . . . . . 59

3.3.2 The polytope $P_k(G, 2)$ . . . . . 61

3.4 Solvability and computational results . . . . . 63

3.5 Concluding remarks . . . . . 67

**4 Rooted Two-Edge Connected Hop-Constrained Network Design Problem** **69**

4.1 Introduction . . . . . . . . . . 69
4.2 Complexity of the THNDP . . . . . . . . . . 71
4.3 Integer programming formulation . . . . . . . . . . 74
4.4 Valid inequalities . . . . . . . . . . 75
4.5 Facets of the rooted THNDP polytope . . . . . . . . . . 83
4.6 Separations . . . . . . . . . . 101
4.7 Branch-and-Cut and computational results . . . . . . . . . . 105
4.8 Concluding remarks . . . . . . . . . . 113

**5 General Two-Edge Connected Hop-Constrained Network Design Problem 115**

5.1 Introduction . . . . . . . . . . 115
5.2 New classes of valid inequalities for $L = 2$ . . . . . . . . . . 116
5.3 New class of valid inequalities for $L = 3$ . . . . . . . . . . 121
5.4 Separation procedures . . . . . . . . . . 123
5.5 Computational results . . . . . . . . . . 124
5.6 Concluding remarks . . . . . . . . . . 128

**6 Two Disjoint 4-Hop-Constrained Paths Problems 131**

6.1 Introduction . . . . . . . . . . 131
6.2 Formulation for the node THPP when $L = 4$ . . . . . . . . . . 132
6.3 Formulation for the edge THPP when $L = 4$ . . . . . . . . . . 136
6.4 Concluding remarks . . . . . . . . . . 144

**Conclusion 147**

# List of Figures


2.1 Support graph of an $L$-$st$-path-cut inequality for $L = 3$ . . . . . . . . . 19
2.2 The auxiliary digraph $D = (N', A)$ of Lemma 2.2.2 . . . . . . . . . . . 21
2.3 An infeasible network for the THPP with $L = 4$ whose incidence vector nevertheless satisfies the candidate formulation . . . . . . . . . . . . . . 23
2.4 The structure generated in Case 2.1 . . . . . . . . . . . . . . . . . . . . . . 42
2.5 The 3-$st$-path-cut generated in Case 2.2 . . . . . . . . . . . . . . . . . 44
2.6 A fractional extreme point for the problem with several demands . . . . 54

3.1 Support graph of a 2-path-cut inequality . . . . . . . . . . . . . . . . . 56
3.2 CPU time in function of $k$ for a given $k$HPP instance when $L = 2$ . . . 66
3.3 CPU time in function of $k$ for a given $k$HPP instance when $L = 3$ . . . 66

4.1 An instance of the dominating set problem (graph $G$) and the corresponding instance of the rooted THNDP with $L = 3$ (graph $\overline{G}$) . . . . . 72
4.2 Support graph of a double cut inequality for $L = 3$ . . . . . . . . . . . 77
4.3 Support graph of a triple path-cut inequality for $L = 2$ . . . . . . . . . 78
4.4 Support graph of a rooted-partition inequality on $p = 5$ demands . . . . 79
4.5 Support graph of a rooted-opt-partition inequality on $p = 5$ demands . 80
4.6 An example of an $(L+2)$-cycle $C$ and the two associated $(L+1)$-cycles $C_1$ and $C_2$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 87
4.7 Best solutions found for 52 cities with 10 demands, on the left, when $L = 2$ (optimal), and, on the right, when $L = 3$ . . . . . . . . . . . . . 111
4.8 Best solutions found for 52 cities with 30 demands, on the left, when $L = 2$ (optimal), and, on the right, when $L = 3$ . . . . . . . . . . . . . 112
4.9 Best solutions found for 52 cities with 55 demands, on the left, when $L = 2$ (optimal), and, on the right, when $L = 3$ . . . . . . . . . . . . . 112

5.1 Support graph of a double path-cut inequality when $L = 2$ ($t_2 \in N \backslash V_1$) 117

5.2 Support graph of an opposed double path-cut inequality . . . . . . . . 118
5.3 Support graph of a type I inequality . . . . . . . . . . . . . . . . . . . . 119
5.4 Support graph of a type II inequality . . . . . . . . . . . . . . . . . . . . 120
5.5 Support graph of a disjoint-partition inequality with $p = 1$ . . . . . . . 121
5.6 Support graph of a double path-cut inequality when $L = 3$ ($t_2 \in N \backslash V_1$) 122

6.1 An infeasible solution to the node THPP for $L = 5$, but satisfying system (6.1)-(6.6) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 136
6.2 An infeasible solution to the edge THPP for $L = 4$, but satisfying inequalities (6.1),(6.3),(6.5),(6.6) . . . . . . . . . . . . . . . . . . . . . . . . 136
6.3 An infeasible solution to the edge THPP for $L = 4$, but satisfying the previous system plus the generalized jump inequalities . . . . . . . . . 137
6.4 Support graph of a two-layered 4-path-cut inequality . . . . . . . . . . 138
6.5 Possible edges in the constructed partition $\Pi$ . . . . . . . . . . . . . . . . 144
6.6 Fractional extreme point for the node THPP with $L = 4$ . . . . . . . . 146

# List of Tables


1.1 Summary of previous and personal works . . . . . . . . . . . . . . . . . 15

3.1 Results for random $k$HPP instances when $L = 2$ and $L = 3$ . . . . . . . 65

3.2 Respective CPU times of the enumerative and cutting plane algoritms for $L = 2, 3$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67

4.1 Results for random instances for $L = 2$ and rooted demands . . . . . . 108

4.2 Results for random instances for $L = 3$ and rooted demands . . . . . . 108

4.3 Results for random instances for $L = 2$ and arbitrary demands . . . . . 109

4.4 Results for random instances for $L = 3$ and arbitrary demands . . . . . 109

4.5 Results for real instances when $L = 2$ . . . . . . . . . . . . . . . . . . . 110

4.6 Results for real instances when $L = 3$ . . . . . . . . . . . . . . . . . . . 110

5.1 Results for random instances for $L = 2$ and arbitrary demands, with basic and disjoint inequalities only . . . . . . . . . . . . . . . . . . . . . 126

5.2 Results for random instances for $L = 3$ and arbitrary demands, with basic and disjoint inequalities only . . . . . . . . . . . . . . . . . . . . . 126

5.3 Results for random instances for $L = 2$ and arbitrary demands, with basic, disjoint and rooted inequalities . . . . . . . . . . . . . . . . . . . 127

5.4 Results for random instances for $L = 3$ and arbitrary demands, with basic, disjoint and rooted inequalities . . . . . . . . . . . . . . . . . . . 128

# Introduction

Now that the telecommunications market is free, the main concerns of Belgacom and its competitors are, without any doubt, reliability, quality of service, and price. However, these objectives are clearly contradictory. To improve the reliability and quality of service in a network often requires increasing the number of its links, and hence, its construction and maintenance costs. These costs must then be recovered by increasing the prices of the services. But, on the contrary, if the network does not appear survivable enough, the company will lose its clients. Therefore, to reach a satisfactory compromise in the design of a telecommunication network, it becomes necessary to fix at first the required survivability, and then to try to minimize its total cost. This is the goal pursued in this thesis.

In all generality, a network design problem consists in determining the cheapest links (connections, cables) to build in order to have a minimum cost network spanning a given set of locations (which may be, for example, computers, cities, or countries) while satisfying certain reliability requirements. Theses locations can be represented by nodes, and all the possible links between them by edges (arcs) between the corresponding nodes. The construction cost of a link can be seen as a weight on the corresponding edge (arc). This yields a weighted (di)graph. The network design problem can then be modelled as finding in this (di)graph a minimum weight subgraph satisfying the required survivability constraints.

In fact, we will consider here a "double" network survivability. First, we will require that the network to be devised must be (Steiner) $k$-edge connected. This kind of "quantitative" reliablity is quite common in network design. It guarantees that the network contains at least $k$ edge-disjoint paths between the pairs of (privileged) nodes. In consequence, the network will stay fully operational, that is, able to route the information between each pair, even after $k-1$ link failures. In practice, since the risk of two simultaneous failures is often low, the 2-edge connectivity already offers a sufficient quantitative reliability for a reasonable cost. Therefore, we will focus most of our study on the 2-edge connected case.

However, a purely quantitative reliability is often considered insufficient nowadays. A minimum cost 2-edge connected network can indeed be made of a unique cycle spanning the $n$ locations, which then obliges the information between two adjacent nodes to travel through the whole cycle if their direct edge fails. The length of the routing path will thus increase from 1 to $n-1$. Clearly, this is unacceptable in certain kinds of networks.

Actually, there are two types of rerouting strategies in telecommunications, the local and the end-to-end reroutings. In the second one, in case of a link failure, the traffic between the terminal nodes is rerouted along alternative paths. These have to be short enough so that this procedure can be accomplished in a minimum amount of time. This strategy is used in the ATM and Internet networks for example. In addition, apart from rerouting needs, the paths cannot be too long in order to guarantee a good quality of service. In data networks, such as Internet, if the route taken by the information is long, it could cause a low transfer speed. For other networks, based on radio waves, the signal itself could be degraded by a long routing if an increasing noise is added each time the signal is received, transformed or reemitted. For all these reasons, $L$-hop-constrained paths, that is, paths of bounded length $L$, offer exactly the kind of additional "qualitative" reliability required. Note that, in practice, the bound $L$ mainly takes values 2, 3 or 4.

In all generality, we will therefore consider the following optimization problem. Given a weighted graph $G = (N, E)$, a set $D \subseteq N \times N$ of pairs of terminals, and two integers $k, L \geq 2$, the *$k$-edge connected $L$-hop-constrained network design problem* consists in finding in $G$ the minimum cost subgraph containing at least $k$ edge-disjoint paths of at most $L$ edges between each pair in $D$. We will study this problem in several particular cases from a polyhedral point of view. We will then extend some of the results obtained to the general problem. Finally, we will make use of these theoretical results in order to conceive an efficient Branch-and-Cut algorithm able to solve real size instances.

More precisely, this thesis will be organized as follows. In Chapter 1, we present some basic definitions and notations in graph and polyhedral theory. We then give an overview of the existing litterature in survivable network design, and explain in more detail why edge connectivity and hop-constrained paths are good requirements in this framework. In Chapter 2, we consider the particular case where $|D| = 1$, $k = 2$ and $L = 2, 3$. We give an integer programming formulation of this problem in the space of the natural design variables. We then show that its linear relaxation is integral, which yields a complete (and minimal) linear description of the associated polytope. We also consider the dominant of this polytope and separation procedures of the introduced inequalities. In Chapter 3, we extend this polyhedral result to any $k \geq 2$ when $L = 2$. We also present some computational results of a cutting plane algorithm in that case. In Chapter 4, we come back to the 2-edge connectivity, but this time several pairs of terminals (demands) are specified in $D$. We first show that the problem is strongly $NP$-hard, for all fixed values of $L$, even when all the demands in $D$ have a node in common. We then proceed by giving several families of inequalities, which are always valid for the problem, but particularly adapted to the rooted case. When $L = 2, 3$, we give necessary and sufficient conditions for these inequalities to be facet-defining and we devise exact and heuristic separation procedures. These are embedded within a Branch-and-Cut algorithm for which we present extensive computational results. Since the inequalities used in it are based on rooted demands, the results appear very good in this situation, but less satisfactory in the case of numerous disjoint demands. Therefore, in Chapter 5, we present additional classes of valid inequalities taking into account the interaction between disjoint demands. We propose some heuristic procedures to separate these inequalities, and we integrate them into our previous Branch-and-Cut algorithm. New computational results are finally presented. In Chapter 6, we consider the most difficult case where $L = 4$. We give integer programming formulations in the space of the natural design variables for two versions of the problem. More precisely, we ask for a minimum cost network containing at least two 4-hop-constrained $st$-paths, either node-disjoint, or edge-disjoint. In the conclusion, we summarize the results presented in this thesis and give some insight regarding future work.

# Chapter 1

# Preliminary Notions and Previous Works

## 1.1 Notions of graph theory

In this section, we present some basic definitions and notations of graph theory, which will be frequently used in the subsequent chapters. For more details, the reader is referred to [9].

In network design problems, the locations and the possible links between them can be represented by an undirected graph $G = (N, E)$. The set $N$, of cardinality $n$, is called the *node set*, and the set $E$, of cardinality $m$, the *edge set*. If $u, v \in N$, we denote by $uv$ a fixed edge between $u$ and $v$. The nodes $u$ and $v$ will be called the *end nodes* of $uv$. The edge $e$ is then *incident* to $u$ and $v$, while these two nodes are *adjacent* to each other. A *loop* in a graph $G$ is an edge whose two end nodes coincide. Two edges are *parallel* if they have the same end nodes. A graph $G = (N, E)$ is *simple* if $E$ does not contain parallel edges or loops. Moreover, the graph $G$ is *complete* if there exists an edge $uv$ between each pair of nodes $u, v \in N$. Implicitly, we will always suppose that the graphs we consider are finite, undirected, and loopless, but neither necessarily simple, nor complete.

We will also consider *directed graphs* (*digraphs*). A directed graph $D$ is a couple $(N, A)$ where $N$ is the *node set* and $A$ is the *arc set*. An *arc* between two nodes

$u, v \in N$ is a couple $(u, v)$ where $u$ is the *tail* of the arc and $v$ its *head*.

Given a digraph $D = (N, A)$ with capacity $u_{ij} \geq 0$ on each arc $(i, j) \in A$, and two nodes $s, t \in N$, an *st-flow* is a set of quantities $\{\phi_{ij}\}$ verifying the following conditions: $0 \leq \phi_{ij} \leq u_{ij}$, for each $(i, j) \in A$, and $\sum_{j=1}^{j=n}(\phi_{ij} - \phi_{ji}) = 0$, for all $i \in N \backslash \{s, t\}$. Moreover, the value of the $st$-flow is given by $c(\phi) = \sum_{j=1}^{j=n}(\phi_{sj} - \phi_{js}) = -\sum_{j=1}^{j=n}(\phi_{tj} - \phi_{jt})$. For $W \subset N$ with $s \in W$ and $t \notin W$, the value $c(W)$ of the *cut* associated to $W$ is equal to $\sum_{i \in W, j \notin W} u_{ij}$. By the Ford and Fulkerson's Theorem, we have that the maximum value of an $st$-flow in $D$ is equal to the minimum value of an $st$-cut.

Let $G = (N, E)$ be an undirected graph. If $W \subset N$ is a node subset of $G$, then the set of edges that have only one node in $W$ is called a *cut* and denoted by $\delta_G(W)$, or simply by $\delta(W)$, when it is obvious that the cut is considered relatively to $G$. We will write $\delta(v)$ for $\delta(\{v\})$. The number of edges incident to some node $v$ is then equal to $|\delta(v)|$, and is called the *degree* of node $v$. Given two nodes $s$ and $t$, a cut $\delta(W)$ such that $s \in W$ and $t \in \overline{W} = N \backslash W$ will be called an *st-cut*. If $V, W \subset N$, $[V, W]$ is the set of edges having one end node in $V$ and the other one in $W$. Note that we will write $[v, w]$ instead of $[\{v\}, \{w\}]$. Also, if $W \subset N$, we will denote by $E(W)$ the set of edges having both end nodes in $W$.

A *partition* $\Pi = (V_0, V_1, \ldots, V_p)$ of $N$ is a collection of disjoint node subsets whose union is $N$. For a partition $\Pi = (V_0, V_1, \ldots, V_p)$ of $N$, the associated *multicut* in $G$, denoted by $\Delta_\Pi(G) = \delta(V_0, V_1, \ldots, V_p)$, is the set of edges having their end nodes in two different subsets. We will denote by $E_\Pi^{q,r} = \bigcup_{i=q,q+1,\ldots,r}[V_i, V_{i+1}]$ the set of edges between the consecutive subsets $V_q, V_{q+1}, \ldots, V_{r+1}$ of $\Pi$. We will call *chord* of the partition an edge that is not between two consecutive subsets of $\Pi$.

If $W \subset N$, we will denote by $G - W$ the subgraph of $G$ obtained by deleting the nodes of $W$ and all the edges having at least one end node in this set. If $W = \{v\}$, we will write $G - v$ for $G - \{v\}$. The subgraph of $G$ *induced* by $W$, that is $G(W) = (W, E(W))$, is then $G - (N \backslash W)$. In the same way, if $F \subset E$, we denote by $G - F$ the subgraph of $G$ obtained by removing the edges in $F$. If $F = \{e\}$, we write $G - e$ instead of $G - \{e\}$.

A *path* $P$ of $G$ is an alternate sequence of nodes and edges $(u_1, e_1, u_2, e_2, \ldots, u_{q-1}, e_{q-1}, u_q)$ where $e_i \in [u_i, u_{i+1}]$ for $i = 1, \ldots, q-1$. We will denote a path $P$ by either its node sequence $(u_1, \ldots, u_q)$ or its edge sequence $(e_1, \ldots, e_{q-1})$. The nodes $u_1$ and $u_q$ are called the *end nodes* of $P$, while its other nodes are said *internal.* A path is *simple* if it does not contain the same node twice. In the sequel, we will always suppose that the paths are simple. By opposition, a non-simple path will be called a *walk.* A path whose end nodes are $s$ and $t$ will be called an *st-path.* A *cycle* in $G$ is a path whose end nodes coincide, that is, $u_1 = u_q$. It will be denoted in the same way as a path. Also, a cycle is simple if, with the exception of $u_1$, it does not contain the same node twice. As for a partition, we call a *chord* an edge between any two non-adjacent nodes of a path (cycle).

Two $st$-paths are *edge-disjoint* if they have no edges in common. They are *node-disjoint* if they have no internal nodes in common. Note that two node-disjoint paths are edge-disjoint, but not conversely. A graph $G$ is *$k$-edge connected* (resp. *$k$-node connected*) if it contains $k$ edge-disjoint (resp. node-disjoint) $st$-paths for all pairs of nodes $\{s,t\} \in N \times N$. We say that $G$ is *Steiner* $k$-edge ($k$-node) connected if it is $k$-edge ($k$-node) connected relative to pairs of privileged nodes in $N$ only. In the sequel, we will omit the qualificative Steiner since we always ask for this reliability over a set $D \subseteq N \times N$, rather than over $N \times N$. The largest integer $k$, such that the graph $G$ is $k$-edge connected (resp. $k$-node connected), is called the *edge connectivity* (resp. *node connectivity*) of $G$. A graph 1-edge connected is also 1-node connected, and is simply said *connected.* A graph that does not contain any cycles is called a *forest.* A connected forest is a *tree.*

Let $G = (N, E)$ be a graph. The *demand set* $D \subseteq N \times N$ is a subset of pairs of nodes, called *demands.* If the pair $\{s,t\}$ is a demand in $D$, we will call $s$ and $t$ *demand nodes* or *terminal nodes.* In particular, when several demands $\{s,t_1\},\ldots,\{s,t_d\}$ are rooted in the same node $s$, we will speak of $s$ as a *source node* and of the $t_i$'s as the *destination nodes* of $s$. The nodes in $N$ that do not belong to any demand of $D$, will be called *Steiner nodes.* Let $L \geq 2$ be a fixed integer. For a demand $\{s,t\} \in D$, an *$L$-st-path* in $G$ is a path between $s$ and $t$ of length at most $L$, where the *length* of a path is defined as the number of its edges (also called *hops*). We will also speak of *$L$-hop-constrained paths.*

## 1.2 Notions of polyhedral theory

In this section, we present the main concepts of polyhedral theory, see [60] for more details.

Given a graph $G = (N, E)$ and an edge subset $F \subseteq E$, the $0-1$ vector $x^F \in I\!R^E$, such that $x^F(e) = 1$ if $e \in F$ and $x^F(e) = 0$ otherwise, is called the *incidence vector* of $F$. The $x$-variables are named *natural design variables* since they are the ones characterizing the topology of the network. The *support graph* of $x^F$ is the subgraph of $G$ containing only the edges $e \in E$ with $x^F(e) = 1$, that is, the edges of $F$. Similarly, the support graph of an inequality $ax \geq \alpha$ is the graph whose edges are such that $a(e) > 0$.

A *linear combination* $x$ of vectors $x_1, \ldots, x_p$ in $I\!R^m$ is $\sum_{i=1}^{i=p} \lambda_i x_i$. If $\sum_{i=1}^{i=p} \lambda_i = 1$, we say that $x$ is an *affine combination* of the $x_i$'s, or, if we have at the same time that every $\lambda_i$ is nonnegative, a *convex combination.* Vectors are *linearly independent* (resp. *affinely independent*) if none of them are a linear (resp. affine) combination of the others. If $S$ is a set of incidence vectors in $I\!R^m$, we will denote by $conv(S)$ the *convex hull* of $S$, that is, the set of convex combinations of the vectors in $S$.

A *polyhedron* $\mathcal{P}$ is the intersection of a finite number of half-spaces in $I\!R^m$. It can also be seen as the set of solutions to a given linear system $Ax \leq b$, where $A \in I\!R^{m' \times m}$ and $b \in I\!R^{m'}$. We then write $\mathcal{P} = \{x \in I\!R^m : Ax \leq b\}$. A polyhedron $\mathcal{P}$ is a *polytope* if it is bounded. The *dimension* of $\mathcal{P}$, $dim(\mathcal{P})$, is the maximum number of affinely independent vectors in $\mathcal{P}$ minus 1. We also have that $dim(\mathcal{P}) = m - rank(A^=)$, where $A^=$ is the submatrix of $A$ of inequalities satisfied with equality by all the vectors in $\mathcal{P}$ (*implicit equalities*). The polyhedron $\mathcal{P}$ is *full dimensional* if $dim(\mathcal{P}) = m$.

An *extreme point* of a polyhedron $\mathcal{P}$ is a vector in $\mathcal{P}$ that cannot be obtained as a convex combination of other vectors in it. A polyhedron whose extreme points are all in $\mathbb{Z}^m$ is *integer.* The *dominant* of a polyhedron $\mathcal{P}$ is the polyhedron $Dom(\mathcal{P})$ containing the vectors $x + y$, where $x \in \mathcal{P}$ and $y \in I\!R^m_+$.

An inequality $ax \leq \alpha$ is *valid* for $\mathcal{P}$ if it is satisfied by all the vectors in $\mathcal{P}$. The set $F_a = \{x \in \mathcal{P} : ax = \alpha\}$ is called the *face of* $\mathcal{P}$ *induced by* $ax \leq \alpha$. A *facet* of $\mathcal{P}$ is a face that is not strictly included in any other face except $\mathcal{P}$ itself. The dimension of a facet is $dim(\mathcal{P}) - 1$. A *complete minimal linear description* of a polyhedron consists of the linear system of its facet-defining inequalities and implicit equalities.

Considering a combinatorial optimization problem, with solution set $\mathcal{S} \subseteq [0,1]^m$,

$$\min\{cx : x \in \mathcal{S}\},$$

is equivalent to considering the linear program

$$\min\{cx : x \in conv(\mathcal{S})\},$$

where the polyhedron $conv(\mathcal{S})$ is given by its complete linear description. However, for most problems, this cannot be obtained explicitly. So, we will rather consider a special *relaxation* of the problem, that is, we will optimize the same objective function on a polyhedron simpler to describe than $conv(\mathcal{S})$, but containing it. If an optimal solution $x^*$ of a relaxation is feasible for the original problem, then it is optimal for it and we are done. If not, we try to generate an inequality valid for $conv(\mathcal{S})$ and violated by $x^*$. An inequality $ax \leq \alpha$ is *violated* by $x^*$ if $ax^* > \alpha$. This step corresponds to what we we call a *separation problem*, and the general procedure for solving linear programs is called *a cutting plane algorithm*, since we iteratively try to cut off infeasible solutions obtained through relaxations by generating additional inequalities.

Given a family of linear inequalities $\mathcal{C}$ and $x^* \in \mathbb{R}^m$, the separation problem associated to $\mathcal{C}$ and $x^*$ consists in verifying if all the inequalities in $\mathcal{C}$ are verified by $x^*$ and, if not, in finding an inequality of $\mathcal{C}$ violated by $x^*$. In practice, this problem will generally be solved exactly when it is polynomial, and heuristically when $NP$-hard. In this latter case, or when all the families describing $conv(\mathcal{S})$ are not known, the cutting plane algorithm can finish with neither a feasible solution, nor a new violated inequality. If this happens, we consider two subproblems of the current relaxation by fixing a fractional component of $x^*$ either to 0 or to 1. This step is called *branching*, and the whole procedure is then *a Branch-and-Cut algorithm*. In each subproblem, we indeed try to obtain a feasible solution by adding additional cuts, and, when there are not any, a new branching is performed.

## 1.3 State of the art in network design

In this section, we present the existing litterature concerning network design from a topological point of view. Apart from the topology, other questions can indeed arise in the conception of a network, such as routing or provisioning. However, in this thesis, we will not consider these further stages of optimization. Moreover, we will focus this overview on works that use a polyhedral approach, since it is the one we have personally adopted.

As explained in the introduction, survivability in network design has become quite an issue in the past years. The first works on this subject considered a purely "quantitative" reliablity, where a survivable network corresponds to a $k$-edge, or $k$-node, connected subgraph. At first, the edge and node connectivity was studied within the framework of graph theory. Until the late eighties, algorithms and heurisitics were only developed for very particular network design problems. For example, the Traveling Salesman Problem (TSP) is nothing but a 2-edge connected network design problem where all the nodes must be of degree 2 [42].

Due to the need of efficient algorithms for the design of survivable telephone networks, Monma and Shallcross [58] developed heuristics finding nearly optimal solutions for problems encountered by the Bell company. At that time, the cutting plane procedure had already shown its efficiency in obtaining optimal solutions, or at least very good bounds, for particular network design problems (like the TSP). Therefore, Grötschel and Monma [36] studied several linear relaxations of survivable models from a polyhedral point of view.

This work was continued by them, along with Stoer, see [37],[38],[39],[40],[41]. They studied edge and node connected subgraphs problems within the framework of a general survivable model and discussed the polyhedral aspects. They developed a Branch-and-Cut algorithm adapted to survivable network design problems. When the index $k$ of connectivity is low (at most 2), their algorithm is very efficient. For higher connectivity (3 or more), the first results on a real instance let hope that problems of this type and size can also be solved efficiently by such an algorithm. One can conclude from their studies that 2-edge (2-node) connected networks are cost effective and provide an adequate level of survivability. In such networks, there are at least two edge-disjoint (node-disjoint) paths between each pair of nodes. So, if a link (a node) fails, it is always possible to reroute the traffic between two terminals along the second path. This explains why the 2-edge (2-node) connected subgraph problem and its associated polytope have been the subject of extensive research in the past years.

In [55], Mahjoub shows that if $G$ is series-parallel then the 2-edge connected subgraph polytope is completely described by the trivial and the cut inequalities. This has been generalized by Baïou and Mahjoub [1] for the Steiner 2-edge connected subgraph polytope and by Didi Biha and Mahjoub [7] for the Steiner $k$-edge connected subgraph polytope for $k$ even. The $k$-edge connected subgraph polytope has also been completely described in the class of series-parallel graphs, see [6]. In [3], Barahona and Mahjoub characterize this polytope when $k = 2$ for the class of Halin graphs, and also consider the node connected case. In [21], Fonlupt and Mahjoub study the fractional extreme points of the linear relaxation of the 2-edge connected subgraph polytope. They introduce an ordering on these extreme points and characterize the minimal extreme points with respect to that ordering. As a consequence, they obtain a characterization of the graph for which the linear relaxation of that polytope is integral. Kerivin, Mahjoub and Nocq [51] describe a general class of valid inequalities for the 2-edge connected subgraph polytope, which generalizes the so-called $F$-partition inequalities [55], and introduce a Branch-and-Cut algorithm for the problem based on these inequalities, the trivial and the cut inequalities. Further work on the 2-edge and 2-node connected subgraph problems can be found in [10],[26],[49].

However, the 2-edge connectivity requirement can be insufficient regarding the reliability of today's telecommunication networks. For example, the optimal solution to the 2-edge connected subgraph problem is often a Hamiltonian cycle (or the union of very few cycles). Therefore, in case of a link failure, the rerouting path between its end nodes will have to go through all the other edges in the cycle. The length of the path used by the information between these two nodes will thus increase from 1 to $n-1$. Clearly, this is not acceptable in certain types of network as explained below.

There are in fact two types of rerouting strategies in telecommunications, the local and the end-to-end reroutings. In the first one, in case of a link failure, the traffic between its end nodes needs to be rerouted. In order to limit the length of this deviation, a solution is, for example, to ask for a network such that each of its links belongs to a bounded cycle. In the end-to-end rerouting strategy, the traffic of each demand pair affected by the failure has to be rerouted along an alternative path. This one has to be short enough so that this procedure can be accomplished in a minimum time. This strategy is used in the ATM and Internet networks, for example. Also, in such situations, the paths cannot be too long in order to guarantee an effective routing of good quality. In data networks, such as Internet, the elongation of the information route could cause a strong loss in the transfer speed. For other networks, the signal itself could be degraded by a long routing. For all these reasons, hop-constrained paths, that is, paths of bounded length, offer exactly the kind of additional "qualitative" reliability required.

In the litterature, bounded paths were first considered within graph theory. For example, relations between the degree of the nodes, the number of edges and the diameter (i.e. the maximum length of the shortest paths for any two nodes) of a graph were derived, see [20] and [57]. Personally, we have also worked on obtaining such relations between graph invariants thanks to a computer-assisted polyhedral method, called *GraPHedron*. In [11], we give, in particular, several classes of optimal linear inequalities between the diameter, the maximum degree and the irregularity of connected graphs.

As already mentioned, another reliability condition was first considered in network design, in order to limit the length of the local reroutings. It requires that each link of the network belongs to a ring (cycle) of bounded length. In [24], Fortz, Labbé and Maffioli consider the 2-node connected subgraph problem with bounded rings. This problem consists in finding a minimum cost 2-node connected subgraph $(N, F)$ such that each edge of $F$ belongs to a cycle of length at most $H$. They describe several classes of facet-defining inequalities for the associated polytope and devise a Branch-and-Cut algorithm for the problem. Fortz and Labbé continue this study in [22] for the special case of unit edge lengths. They present a new formulation of the problem and derive facet results for different classes of valid inequalities, which are then integrated into their Branch-and-Cut algorithm. Additional results are presented in [23]. In [25], Fortz, Mahjoub, McCormick and Pesneau study the edge connected version of that problem. They give an integer programming formulation in the space of the natural design variables, and describe different classes of valid inequalities. They study the separation problem for these inequalities and discuss a Branch-and-Cut algorithm.

It remains that $H$-bounded cycles can still not be sufficient in many cases, in particular, when the end-to-end rerouting strategy is used. In fact, in the case of a link failure, we only have the guarantee that the alternative path to route the information between two nodes will increase of at most $H - 2$. In practice, instead of covering the network by cycles $\leq H$, we would rather have two available paths of equivalent length, that is, $L = H/2$. This is the reason why we choose to consider in this thesis this kind of reliability, along with the $k$-edge connectivity, with $k \geq 2$.

To the best of our knowledge, hop-constrained paths, for the design of survivable networks, were first suggested by Leblanc and Reddoch [52]. Since then, research has concerned this requirement along with the 1-connectivity, which is not very satisfactory as quantitative reliability in telecom network design. In [19], Dahl and Johannessen consider the 2-path network design problem, which consists in finding a minimum cost subgraph connecting each pair of terminal nodes by at least one path of length at most 2. This problem is NP-hard. Dahl and Johannessen give an integer programming formulation for the problem and describe some classes of valid inequalities. Using these, they devise a cutting plane algorithm and present some computational results.

The closely related problem of finding a minimum cost spanning tree with hop-constraints is considered in [30],[31],[34]. Here, the hop-constraints limit the number of links between the root and any terminal in the network to a positive integer $L$. This problem is NP-complete, even for $L = 2$. Gouveia [30] gives a multicommodity flow formulation for that problem and discusses a Lagrangean relaxation improving the LP bound. Gouveia [31] and Gouveia and Requejo [34] propose more efficient Lagrangean based schemes for the problem and its Steiner version. Dahl [13] studies the problem for $L = 2$ from a polyhedral point of view and gives a complete description of the associated polytope when the graph is a wheel. Gouveia and Janssen [32] discuss a generalized problem where cables with different reliabilities are available. They formulate the problem as a directed multicommodity flow model and use Lagrangean relaxation together with subgradient optimization to derive lower bounds. Gouveia and Magnanti [33] consider the problem that consists in finding a minimum spanning tree such that the number of edges in the tree between any pair of nodes is limited to a given bound (diameter). They present directed and undirected multicommodity formulations along with some computational experiments. Further hop-constrained survivable network design problems are studied in [2],[4],[5],[53],[59]. A survey of survivability with hop-constraints can be found in [50].

In [14], Dahl considers the hop-constrained path problem, that is the problem of finding between two distinguished nodes $s$ and $t$ a minimum cost path with no more than $L$ edges when $L$ is fixed. He gives a complete description of the dominant of the associated polytope when $L \leq 3$. In the framework of the minimum cost spanning tree problem with hop-constraints, Dahl and Gouveia [17] (see also [16]) consider the directed hop-constrained path problem. They describe various classes of valid inequalities and show that some of these are sufficient to completely describe the associated polytope when $L \leq 3$. Then they discuss some applications to the hop-constrained minimum spanning tree problem. In [12], Coullard, Gamble and Liu investigate the structure of the polyhedron associated with the $st$-walks of length $L$ of a graph, where a walk is a path that may go through the same node more than once. They present an extended formulation of the problem and, using projection, they give a linear description of the associated polyhedron. They also discuss classes of facets of that polyhedron. Dahl et al. [15] also consider the hop-constrained walk polytope and characterize it for $L = 4$. One of their conclusions is that the structure of the hop-constrained path polytope for $L = 4$ is a lot more complicated.

Itai, Perl and Shiloach [48] study the complexity of several variants of the maximum disjoint hop-constrained paths problem. This consists in finding the maximum number of disjoint paths between two nodes $s$ and $t$ of length equal to (or bounded by) $L$, where $L$ is a positive integer. They show that the problem is NP-complete for $L \geq 5$ and polynomially solvable for some of the variants for $L \leq 4$. In particular, they devise a polynomial time algorithm for the problem when the paths must be node- (resp. edge-) disjoint and of length bounded by $L$, with $L \leq 4$ (resp. $L \leq 3$). Bley [8] addresses approximation and computational issues for the edge- (node-) disjoint hop-constrained paths problem. In particular, he shows that the problem of computing the maximum number of edge-disjoint paths between two given nodes of length equal to 3 is polynomial. This answers an open question in [48]. In [54], Li, McCormick and Simchi-Levi study the problem of finding $K$ disjoint paths of minimum total cost between two distinguished nodes $s$ and $t$, where each edge of the graph has $K$ different costs and the $j$th edge-cost is associated with the $j$th path. They show that all the variants of the problem, when the graph is directed or undirected and the paths are edge- or node-disjoint, are NP-complete, even when $K = 2$.

In this thesis, we consider the *k-edge connected L-hop-constrained network design problem*. Given a weighted graph $G = (N, E)$, a set of pairs of terminal nodes $D \subseteq N \times N$, and two integers $k, L \geq 2$, it consists in finding in $G$ the minimum cost subgraph containing at least $k$ edge-disjoint paths of at most $L$ hops between each pair of $D$. Despite its great interest in terms of survivability, this network design problem has never been considered in the existing litterature. In the next chapters, we will present the results that we have obtained concerning it, in function of $k$, $L$ and $D$.

Table 1.1 summarizes where our results are situated respectively to some of the previous works presented in this section. Of course, we do not claim that this list is complete. Moreover, when a same article deals with several variants (as for example the edge and node connected cases), we reference it only once in the cell that we personally find the most relevant. When a chapter is mentioned in parentheses in a cell, it means that, although it does not mainly concern the corresponding case, it nevertheless evoke similar results for it. Note that we will also briefly consider the node version of the $k$-edge connected $L$-hop-constrained network design problem, that is, when the $k$ paths of length $\leq L$ must be node-disjoint rather than edge-disjoint.

Table 1.1 is organized as follows. The first line of the table corresponds to the 1-connectivity case. In the subsequent lines, increasing quantitative reliability conditions are considered, namely the $k$-edge connectivity and the $k$-node connectivity, first for $k = 2$, then for $k \geq 3$. The first column concerns works where no additional qualitative reliability is required. In the second one, the networks must be covered by cycles of bounded length. In the last four columns, we ask for hop-constrained paths. Several variants can be studied, namely for a single demand pair and $L = 2/L \geq 4$, and for multiple demand pairs, either rooted (i.e. having a node in common) or arbitrary. Observe that our work deals with this last part of the table, except for higher $k$ connectivity ($k \geq 3$) and several demands ($|D| \geq 2$) at the same time.

Table 1.1: Summary of previous and personal works

| | | No qual. r. | $H$-cycles | $L$-$st$-paths ($\|D\| = 1$) | | $L$-$st$-paths ($\|D\| \geq 2$) | |
|---|---|---|---|---|---|---|---|
| | | | | $L = 2, 3$ | $L \geq 4$ | $D$ rooted | $D$ arbitrary |
| | $k = 1$ | many works | NA | [14] | [15],[16] | [13],[17],[30],[31],[34] | [19] |
| $k$-edge c. | $k = 2$ | [1],[21],[51],[55] | [25] | Chapter 2 | Chapter 6 | Chapter(s) 4 (6) | Chapter(s) 5 (6) |
| | $k \geq 3$ | [6],[7] | ? | Chapter 3 | ? | ? | ? |
| $k$-node c. | $k = 2$ | [3],[38],[39] | [22],[23],[24] | (Chapters 2/6) | Chapter 6 | (Chapter 6) | (Chapter 6) |
| | $k \geq 3$ | [36],[41] | ? | (Chapter 3) | [8],[48],[54] | ? | ? |

In Chapter 2, we will first study the $k$-edge connected $L$-hop-constrained network design problem in the particular case where $|D| = 1$, $k = 2$ and $L = 2, 3$. In the concluding remarks (Section 2.8), we will also briefly consider the node version of this specific problem. We will then study more general cases, namely when $k \geq 3$ (Chapter 3), when $|D| \geq 2$ (Chapters 4 and 5), and when $L \geq 4$ (Chapter 6).

# Chapter 2

# Two Edge-Disjoint Hop-Constrained Paths Problem

In this chapter, we consider the $k$-edge connected $L$-hop-constrained network design problem in the particular case where the demand set $D$ is reduced to a single pair $\{s, t\}$, and $k = 2$. Moreover, our main results will hold for $L = 2, 3$ only. We call this problem the *Two edge-disjoint Hop-constrained Paths Problem* (THPP for short). This study was realized with A. Ridha Mahjoub and Pierre Pesneau. It has been the subject of an article published in SIAM Journal on Discrete Mathematics [47].

## 2.1 Introduction

Given a graph $G = (N, E)$ with distinguished nodes $s$ and $t$, and a fixed integer $L \geq 2$, an *$L$-$st$-path* in $G$ is a path between $s$ and $t$ of length at most $L$, where the length of a path is the number of its edges. Given a function $c : E \rightarrow \mathbb{R}$ which associates a cost $c(e)$ to each edge $e \in E$, the *Two edge-disjoint Hop-constrained Paths Problem* (THPP) is to find a minimum cost subgraph such that between $s$ and $t$ there exist at least two edge-disjoint $L$-$st$-paths.

It is clear that, when $L$ is fixed, an optimal solution of the THPP can be computed in polynomial time by enumerating all the $L$-$st$-paths (given that there is a polynomial number of them). However, in a complete graph $G = (N, E)$ with $|N| = n$, there are

$\mathcal{O}(n^{L-1})$ $L$-$st$-paths, which can also be enumeratively generated in $\mathcal{O}(n^{L-1})$ time. For every pair of such paths, one has to verify their edge-disjunction, which requires $\mathcal{O}(L^2)$ comparisons. Consequently, the whole enumerative algorithm for the THPP runs in $\mathcal{O}(L^2 n^{2(L-1)})$ time. Clearly, such a method is far from being applicable in practice (see also Section 3.4). This is even worse for non simple graphs since there can then be a non polynomial number of $L$-$st$-paths to enumerate. One of the principal aims of this work is to devise a more efficient algorithm for the THPP. This algorithm, which will be a cutting plane method, will be based on a complete description of the associated polytope by a system of linear inequalities.

The THPP can also be seen as a special case of our general problem, where more than one pair of terminals is considered. Thus an efficient algorithm for solving the THPP would be useful to solve the $k$-edge connected $L$-hop-constrained network design problem. In the sequel, we will try to extend the results obtained here first to any $k \geq 2$ (Chapter 3), then to multiple demands in $D$ (Chapters 4 and 5), and finally, to greater values of $L$ (Chapter 6). Our strongest results here will indeed concern $L = 2, 3$.

Given a graph $G = (N, E)$ and an edge subset $F \subseteq E$, the $0-1$ vector $x^F \in I\!R^E$, such that $x^F(e) = 1$ if $e \in F$ and $x^F(e) = 0$ otherwise, is called the *incidence vector* of $F$. For $L \geq 2$, the *convex hull* of the incidence vectors of the solutions of the THPP on $G$, denoted by $P(G, L)$, will be called the THPP *polytope*. If $W \subset N$ is a node subset of $G$, then the set of edges that have only one node in $W$ is called a *cut* and denoted by $\delta(W)$. We will write $\delta(v)$ for $\delta(\{v\})$. A cut $\delta(W)$ such that $s \in W$ and $t \in \overline{W} = N \backslash W$ will be called an *st-cut*. For more definitions and notations, the reader is referred to Chapter 1.

If $x^F$ is the incidence vector of the edge set $F$ of a solution of the THPP, then $x^F$ satisfies the following inequalities

$$x(\delta(W)) \geq 2, \quad \text{for all } st\text{-cut } \delta(W), \tag{2.1}$$

$$1 \geq x(e) \geq 0, \quad \text{for all } e \in E. \tag{2.2}$$

Inequalities (2.1) will be called *st-cut inequalities* and inequalities (2.2) *trivial inequalities*. Their respective validity for the THPP will be proved in details in Section 2.2.

In [14], Dahl considers the problem of finding a minimum cost path between two given terminal nodes $s$ and $t$ of length at most $L$. He describes a class of valid inequalities for the problem and gives a complete description of the dominant of the associated $L$-$st$-path polyhedron when $L \leq 3$. In particular, he introduces a class of valid inequalities as follows.

Let $V_0$, $V_1$, ..., $V_{L+1}$ be a partition of $N$ such that $s \in V_0$, $t \in V_{L+1}$ and $V_i \neq \emptyset$ for all $i \in \{1, \ldots, L\}$. Let $T$ be the set of edges $e = uv$ where $u \in V_i$, $v \in V_j$ and $|i - j| > 1$. Then the inequality

$$x(T) \geq 1$$

is valid for the $L$-$st$-path polyhedron.

Using the same partition, this inequality can be generalized in a straightforward way to the THPP polytope as

$$x(T) \geq 2. \tag{2.3}$$

The set $T$ is called an *L-(st-)path-cut* and a constraint of type (2.3) an *L-(st-)path-cut inequality*. See Figure 2.1 for an example of this structure for $L = 3$. Again, we will show their validity in the next section.

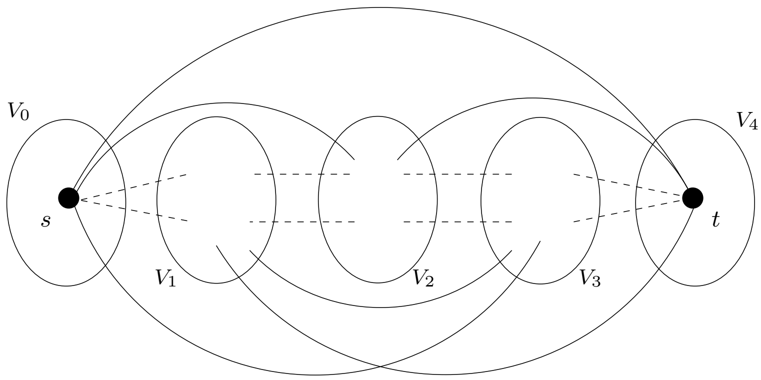


Figure 2.1: Support graph of an $L$-$st$-path-cut inequality for $L = 3$

Let $Q(G, L)$ be the solution set of the system given by inequalities (2.1)-(2.3). In this chapter, we show that inequalities (2.1)-(2.3), together with the integrality constraints, give an integer programming formulation of the THPP for $L = 2, 3$. We then discuss the THPP polytope, $P(G, L)$, and show that $P(G, L) = Q(G, L)$ when $L = 2, 3$ for any graph. Since inequalities (2.1),(2.3) can be separated in polynomial time when $L = 2, 3$, this yields a polynomial time cutting plane algorithm for the THPP in that case. We also give necessary and sufficient conditions for inequalities (2.1)-(2.3) to define facets for any $L \geq 2$ when the graph is complete. We finally investigate the dominant of $P(G,L)$, for which we give a complete description for any $L \geq 2$ when $P(G,L) = Q(G,L)$. As a consequence, we obtain the dominant of $P(G,L)$ when $L = 2, 3$.

The chapter is organized as follows. In the next section, we give an integer programming formulation of the THPP for $L \leq 3$. In Section 2.3, we study the THPP polytope when $L = 2, 3$ and give our main result. In Section 2.4, we study some structural properties of the facet-defining inequalities of $P(G,L)$, which are used in Section 2.5 for proving our main result. In Section 2.6, we describe necessary and sufficient conditions for the inequalities (2.1)-(2.3) to be facet-defining. In Section 2.7, we discuss the dominant of $P(G,L)$ and, in Section 2.8, we give some concluding remarks.

## 2.2 Formulation for $L = 2, 3$

First, we show that inequalities (2.1)-(2.3) are valid for the THPP for any $L \geq 2$.

**Theorem 2.2.1.** *For any $L \geq 2$, inequalities (2.1)-(2.3) are valid for $P(G,L)$.*

*Proof.* We will show that any feasible solution $(N, F)$ to the THPP for any $L \geq 2$ has an incidence vector $x^F$ satisfying constraints (2.1)-(2.3).
Let $G_F$ be the support graph of $x^F$. First, it is clear that the constraints $0 \leq x^F(e) \leq 1$ are all satisfied since $x^F$ is a boolean vector. Suppose now that there exists a subset of nodes $W$, containing $s$ and not $t$, such that $x^F(\delta(W)) \leq 1$. Then there is at most one edge $e \in \delta(W)$ such that $x^F(e) = 1$, and all the $st$-paths in $G_F$ must go through $e$. This contradicts the existence of two edge-disjoint $st$-paths in $G_F$. Finally, if there exists an $L$-$st$-path-cut $T$ such that $x^F(T) \leq 1$, there is at most one edge $e \in T$ with $x^F(e) = 1$. Therefore, if one $st$-path of $G_F$ uses $e$, then any other can only go through edges of $F \backslash T$. Clearly, the minimum number of hops of such a path is $L + 1$, which also contradicts the feasibility of the solution. □

We are now going to show that the $st$-cut, $L$-$st$-path-cut and trivial inequalities, together with the integrality constraints, suffice to formulate the THPP as a $0-1$ linear program when $L = 2, 3$. To this end, we first give a lemma.

**Lemma 2.2.2.** *Let $G = (N, E)$ be a graph, $s$, $t$ two nodes of $N$ and $L \in \{2, 3\}$. Suppose that there do not exist two edge-disjoint $L$-$st$-paths in $G$. Then all the $L$-$st$-paths in $G$ (if any) have one edge in common.*

*Proof.* We first show the statement for $L = 3$. The proof uses ideas from [25] and [48]. Consider the capacitated directed graph $D = (N', A)$ obtained from $G$ in the following way. The set $N'$ consists of copies $s', t'$ of $s, t$ and two copies $N_1, N_2$ of $N \backslash \{s, t\}$. For $u \in N \backslash \{s, t\}$, let $u_1$ and $u_2$ be the corresponding nodes in $N_1$ and $N_2$, respectively. To each edge $e \in [s, u]$, with $u \in N \backslash \{s, t\}$, we associate an arc $e'$ from $s'$ to $u_1$ of capacity 1. To each edge $e \in [v, t]$, with $v \in N \backslash \{s, t\}$, we associate an arc $e'$ from $v_2$ to $t'$ of capacity 1. For an edge $e \in [u, v]$, with $u, v \in N \backslash \{s, t\}$, we consider two arcs, one from $u_1$ to $v_2$ and the other from $v_1$ to $u_2$, both of capacity 1. Finally, we consider in $D$ an arc from $s'$ to $t'$ of capacity 1 for every edge in $[s, t]$, and an arc from each node of $N_1$ to its peer in $N_2$ with infinite capacity (see Fig. 2.2 for an illustration). Note that multiple edges in $G$ yield multiple arcs in $D$.

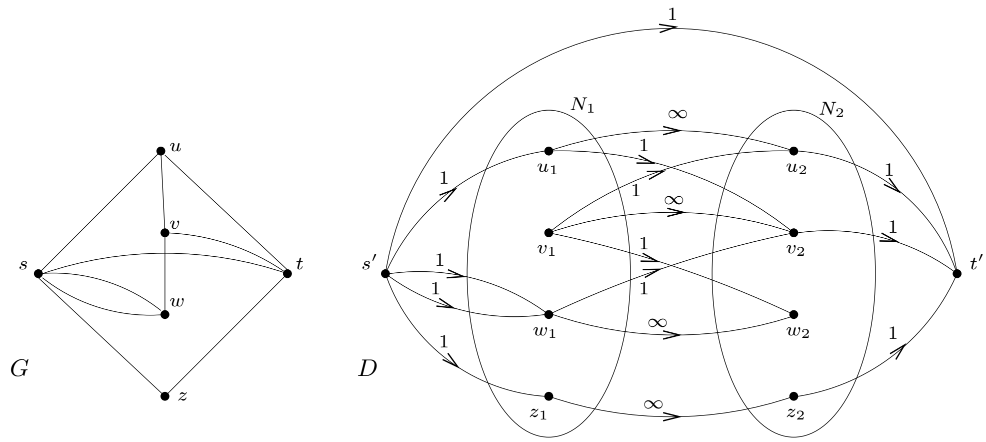


Figure 2.2: The auxiliary digraph $D = (N', A)$ of Lemma 2.2.2

Observe that there is a one-to-one correspondance between the 3-$st$-paths in $G$ and the directed $s't'$-paths in $D$.

Now consider a maximum flow $\phi \in I\!R_+^A$ from $s'$ to $t'$ in $D$. As the capacities of $D$ are integer, $\phi$ can be supposed to be integer. Hence the flow value of each arc of capacity 1 is either 0 or 1. We claim that $\phi$ can be chosen so that no two arcs $(u_1, v_2)$ and $(v_1, u_2)$, corresponding to the same edge $uv$ in $G$, have a positive value. Indeed, suppose that $\phi(u_1, v_2) = 1$ and $\phi(v_1, u_2) = 1$. Let $\phi' \in I\!R_+^A$ be the flow given by

$$\phi'(e) = \begin{cases} \phi(e) + 1 & \text{if } e \in \{(u_1, u_2), (v_1, v_2)\}, \\ 0 & \text{if } e \in \{(u_1, v_2), (v_1, u_2)\}, \\ \phi(e) & \text{otherwise.} \end{cases}$$

As $(u_1, u_2)$ and $(v_1, v_2)$ have infinite capacity and the flow going into $u_2$ and $v_2$ has not changed, $\phi'$ is still feasible. Moreover, $\phi'$ has the same value as $\phi$.

As a consequence, an $s't'$-flow of value $q$ in $D$ corresponds to $q$ edge-disjoint 3-$st$-paths in $G$. Since there do not exist, in $G$, two edge-disjoint 3-$st$-paths, the maximum flow in $D$ is of value at most one. Hence a minimum $st$-cut in $D$ is of value at most 1 as well. Observe that such a cut does not contain arcs with infinite capacity. Hence, a minimum cut corresponds to a set of one edge that intersects all the 3-$st$-paths of $G$, and the proof for $L = 3$ is complete.

If $L = 2$, then we can similarly show the statement by considering the digraph $D = (N', A)$ where $N'$ is a copy of $N$ and to every edge $e \in [s, u]$ (resp. $[u, t]$, resp. $[s, t]$), where $u \in N \backslash \{s, t\}$, corresponds an arc $e'$ from $s'$ to $u'$ (resp. $u'$ to $t'$, resp. $s'$ to $t'$) of capacity 1 in $D$. Here $u'$ is the copy of $u$ in $N'$ for every $u \in N$. □

**Theorem 2.2.3.** *Let $G = (N, E)$ be a graph and $L \in \{2, 3\}$. Then the THPP is equivalent to the integer program*

$$\mathit{Min}\ \{cx\ :\ x \in Q(G, L),\ x \in \mathbb{Z}^E\}.$$

*Proof.* To prove the theorem, it is sufficient to show that every $0 - 1$ solution $x$ of $Q(G, L)$ induces a solution of the THPP. Let us assume the contrary. Suppose that $x$ does not induce a solution of the THPP but satisfies the $st$-cut and trivial constraints. We will show that $x$ necessarily violates at least one of the $L$-$st$-path-cut constraints $x(T) \geq 2$. Let $G_x$ be the subgraph induced by $x$. As $x$ is not a solution of the problem, $G_x$ does not contain two edge-disjoint $L$-$st$-paths. As $L \in \{2, 3\}$, it follows, by Lemma 2.2.2, that there exists at most one edge in $G_x$ that intersects every $L$-$st$-path. Consider the graph $\tilde{G}_x$ obtained from $G_x$ by deleting this edge. Obviously, $\tilde{G}_x$ does not contain any $L$-$st$-path.

We claim that $\tilde{G}_x$ contains at least one $st$-path of length at least $L + 1$. In fact, as $x$ is a $0 - 1$ solution and satisfies the $st$-cut inequalities, $G_x$ contains at least two edge-disjoint $st$-paths. Since at most one edge was removed from $G_x$, at least one path remains between $s$ and $t$ in $\tilde{G}_x$. However, since $\tilde{G}_x$ does not contain an $L$-$st$-path, that path must be of length at least $L + 1$.

Now consider the partition $V_0, \ldots, V_{L+1}$ of $N$, with $V_0 = \{s\}$, $V_i$ the set of nodes at distance $i$ from $s$ in $\tilde{G}_x$, for $i = 1, \ldots, L$, and $V_{L+1} = N \setminus \left(\bigcup_{i=0}^{L} V_i\right)$, where the distance between two nodes is the length of a shortest path between these nodes. Since there does not exist an $L$-$st$-path in $\tilde{G}_x$, it is clear that $t \in V_{L+1}$. Moreover, as, by the claim above, $\tilde{G}_x$ contains an $st$-path of length at least $L + 1$, the sets $V_1, \ldots, V_L$ are nonempty. Furthermore, no edge of $\tilde{G}_x$ is a chord of the partition (that is, an edge between two sets $V_i$ and $V_j$, where $|i - j| > 1$). In fact, suppose that there exists an edge $e = v_i v_j \in [V_i, V_j]$ with $|i - j| > 1$ and $i < j$. Therefore $v_j$ is at distance $i + 1$ from $s$, a contradiction.

Thus, the edge deleted from $G_x$ is the only edge that may be a chord of the partition in $G_x$. In consequence, if $T$ is the set of chords of the partition in $G$, then $x(T) \leq 1$. But this implies that the corresponding $L$-$st$-path-cut inequality is violated by $x$. □

If $L \geq 4$, inequalities (2.1)-(2.3), together with the integrality constraints $x(e) \in \{0, 1\}$, for all $e \in E$, do not suffice to formulate the THPP as an integer program. Indeed, suppose that $L = 4$ and consider the graph shown in Fig. 2.3.

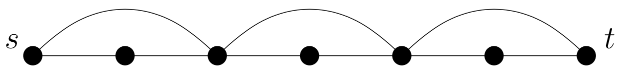


Figure 2.3: An infeasible network for the THPP with $L = 4$ whose incidence vector nevertheless satisfies the candidate formulation

It is not hard to see that the solution induced by this graph satisfies inequalities (2.1)-(2.3) whereas the graph itself is not a feasible solution of the THPP.

For the moment, we will let this as an open question. The already curious reader is referred to Chapter 6, where we propose an integer programming formulation for the THPP when $L = 4$ (and also for its node version).

The separation problem for a system of inequalities consists in verifying whether a given solution $x^* \in I\!R^E$ satisfies the system and, if not, in finding an inequality of the system that is violated by $x^*$. The separation problem for inequalities (2.1) can be solved in polynomial time using any polynomial max-flow algorithm (e.g. [44]). Inequalities (2.3) can also be separated in polynomial time when $L \leq 3$. In fact, in this case, it is not hard to see that the separation problem reduces to finding a minimum weight edge subset that intersects all $L$-$st$-paths. Recently, Fortz et al. [25] have shown that this problem reduces to a max-flow problem (as described in the proof of Lemma 2.2.2) and hence can be solved in polynomial time.

Thus, by the ellipsoid method [35], the THPP can be solved in polynomial time. It would then be interesting to characterize the graphs for which $Q(G, L)$ is integral. In what follows, we will show that $Q(G, L)$ is integral for any graph when $L = 2, 3$.

## 2.3 THPP polytope for $L = 2, 3$

We first state the main result of this chapter.

**Theorem 2.3.1.** *$P(G, L) = Q(G, L)$, if $L = 2, 3$.*

The proof of this theorem will be given in Section 2.5. In what follows, we shall discuss the dimension of $P(G, L)$ and study some properties of its facial structure. Let $G = (N, E)$ be a graph. An edge $e \in E$ will be called *$L$-$st$-essential* if $e$ belongs to an $st$-cut of cardinality 2 or an $L$-$st$-path-cut of cardinality 2. Let $E^*$ denote the set of $L$-$st$-essential edges. Thus, $P(G - e, L) = \emptyset$ for all $e \in E^*$. The following theorem characterizes the dimension of the polytope $P(G, L)$.

**Theorem 2.3.2.** $\dim(P(G, L)) = |E| - |E^*|$.

*Proof.* Let $e$ be an $L$-$st$-essential edge of $G$, that is, $e \in E^*$. Then, by definition, there exists an $st$-cut $\delta(W)$ (resp. an $L$-$st$-path-cut $T$) of cardinality two containing $e$. Since every solution to the THPP must contain at least two edges from $\delta(W)$ (resp. from $T$), we have in fact that it contains both edges of $\delta(W)$ (resp. of $T$), and hence $e$. Therefore, the incidence vector of any solution to the THPP satisfies $x(e) \geq 1$ as an equation. This implies that the dimension of $P(G, L)$ is at most $|E| - |E^*|$.
On the other hand, for any edge $e$ that is not $L$-$st$-essential, we have that $E \backslash \{e\}$ is a feasible solution to the THPP. Moreover, the edge sets $E$ and $E \backslash \{e\}$, for every $e \in E \backslash E^*$, are clearly affinely independent. Since there are $|E| - |E^*| + 1$ such solutions, we obtain that $dim(P(G, L)) \geq |E| - |E^*|$, and the result is proved. □

**Corollary 2.3.3.** *If $G = (N, E)$ is complete with $|N| \geq 4$, then $P(G, L)$ is full dimensional.*

The following theorem gives a procedure for obtaining a linear description of the THPP polytope for a subgraph of $G$ from that corresponding to $G$.

**Theorem 2.3.4.** *Let $G = (N, E)$ be a graph, $s, t$ two nodes of $N$ and $L \geq 2$ an integer. Let $e$ be an edge of $E$. Let $G' = (N, E')$ be the graph obtained from $G$ by deleting $e$. Then a linear system describing $P(G', L)$ can be obtained from a system describing $P(G, L)$ by removing the variables corresponding to $e$.*

*Proof.* Suppose, on the contrary, that there is a facet-defining inequality $a'x \geq \alpha$ of $P(G', L)$ which does not come from a facet-defining inequality of $P(G, L)$. Then any valid inequality of $P(G, L)$ of the form $ax = a'x + \sigma x(e) \geq \alpha$, with $\sigma \in \mathbb{R}$, does not define a facet of $P(G, L)$. Thus $ax \geq \alpha$ is dominated by a linear combination of facet-defining inequalities of $P(G, L)$. That is there are valid constraints $b_i x \geq \beta_i$, $i = 1, \ldots, p$, defining facets of $P(G, L)$ and $\lambda_1, \ldots, \lambda_p > 0$, such that

$$a \geq \sum_{i=1}^{p} \lambda_i b_i, \qquad \alpha \leq \sum_{i=1}^{p} \lambda_i \beta_i.$$

Let $r = |\{i : b'_i \neq 0\}|$. Here $b'_i$ is the restriction of $b_i$ on $E'$. Note that, from the hypothesis, it follows that $r \geq 2$. As $b'_i x \geq \beta_i$ is valid for $P(G', L)$ for $i = 1, \ldots, p$, we obtain that $\sum_{i=1}^{p} \lambda_i b'_i x \geq \sum_{i=1}^{p} \lambda_i \beta_i$ is valid for $P(G', L)$. Now observe that this latter inequality dominates $a'x \geq \alpha$, which contradicts the fact that this inequality defines a facet of $P(G', L)$. □

In the following, we will suppose that $G = (N, E)$ is complete with $|N| \geq 4$. Hence, by Corollary 2.3.3, $P(G, L)$ is full dimensional. If $G = (N, E)$ is not complete, then a description of $P(G, L)$ can be obtained from that of $P(\overline{G}, L)$, using repeatedly Theorem 2.3.4. Here $\overline{G}$ is the complete graph obtained from $G$ by adding the missing edges. Moreover, it is clear that the problem can be reduced to that case by associating a big cost to the missing edges in the graph.

Let

$$S(G) = \{F \subseteq E \,|\, (N, F) \text{ is a solution of the THPP}\}.$$

Given an inequality $ax \geq \alpha$ that defines a facet of $P(G, L)$, we let

$$S_a(G) = \{F \in S(G) \,|\, ax^F = \alpha\}.$$

In what follows, we will consider $a(e)$ as a weight on $e$. Hence, any solution $S$ of $S_a(G)$ will have a weight $a(S)$ equal to $\alpha$, and any solution of $S(G)$ a weight $\geq \alpha$.

**Lemma 2.3.5.** *i) Let $ax \geq \alpha$ be a facet-defining inequality of $P(G, L)$, different from the trivial inequalities. Then for every edge $e \in E$, there exists an edge subset in $S_a(G)$ that contains $e$ and another one that does not.*
*ii) Let $ax \geq \alpha$ be a facet-defining inequality of $P(G, L)$, different from the st-cut inequalities. Then for every st-cut $\delta(W)$, there exists an edge subset in $S_a(G)$ containing at least three edges of $\delta(W)$.*

*Proof.* i) Let $ax \geq \alpha$ be a facet-defining inequality of $P(G, L)$ different from a trivial inequality, and let $e \in E$. Suppose that all edge sets $F \in S_a(G)$ do not contain $e$ (resp. contain $e$). But then their incidence vector $x^F$ satisfies $x(e) \geq 0$ (resp. $x(e) \leq 1$) as equation, a contradiction.
ii) Let $ax \geq \alpha$ be a facet-defining inequality of $P(G, L)$ different from an $st$-cut inequality, and let $\delta(W)$ be an $st$-cut. Suppose that all edge sets $F \in S_a(G)$ contain exactly two edges from $\delta(W)$. Therefore, their incidence vector $x^F$ satisfies $x(\delta(W)) \geq 2$ with equality, which is a contradiction.

□

Lemma 2.3.5 will be frequently used in the sequel. At times we will use it without referring to it explicitely.

**Lemma 2.3.6.** *Let $ax \geq \alpha$ be a facet-defining inequality of $P(G, L)$, different from a trivial inequality. Then $a(e) \geq 0$, for all $e \in E$ and $\alpha > 0$.*

*Proof.* Assume, on the contrary, that there is an edge $e \in E$ such that $a(e) < 0$. Since $ax \geq \alpha$ is different from $-x(e) \geq -1$, by Lemma 2.3.5 i), there must exist a solution $S$ of $S_a(G)$ that does not contain $e$. As $S' = S \cup \{e\}$ still belongs to $S(G)$, this yields $\alpha \leq ax^{S'} = ax^S + a(e) < ax^S = \alpha$, a contradiction. Thus, $a(e) \geq 0$ for all $e \in E$. Since $ax \geq \alpha$ defines a facet of $P(G, L)$, there must exist at least one edge, say $f$, with $a(f) > 0$. Now, as $ax \geq \alpha$ is different from the inequality $x(f) \geq 0$, there is an edge set of $S_a(G)$ containing $f$. This implies that $\alpha > 0$. □

The following lemma shows that parallel edges in $G$ have the same coefficient in every non trivial facet-defining inequality of $P(G, L)$ for $L = 2, 3$.

**Lemma 2.3.7.** *Let $L = 2, 3$ and $ax \geq \alpha$ be a facet-defining inequality of $P(G, L)$ different from the trivial inequalities. Let $[u, v] = \{e_1, e_2, \ldots, e_p\}$ be the set of the parallel edges between two nodes $u$ and $v$ in $G$. Then $a(e_i) = a(e_j)$, for $i, j = 1, \ldots, p$.*

*Proof.* We will show the result for $L = 3$. The proof for $L = 2$ is similar. First we show that all edges in $[u, v]$ have the same coefficient, except possibly one, that may have a smaller coefficient. Indeed, if there are three edges $e_1, e_2, e_3 \in [u, v]$ such that $a(e_1) > a(e_2) \geq a(e_3)$, then there cannot exist an edge subset of $S_a(G)$ containing $e_1$. For otherwise, one could replace $e_1$ by either $e_2$ or $e_3$ and get a solution which violates $ax \geq \alpha$, a contradiction. Now, suppose that there are two edges $e_1, e_2 \in [u, v]$ such that $a(e_1) > a(e_2)$. By the remark above, it follows that $a(e) = a(e_1)$, for all $e \in [u, v] \setminus \{e_1, e_2\}$.

*Claim 1.* Let $S$ be a solution of $S_a(G)$.
i) If $S$ contains $e_1$, then it must contain $e_2$.
ii) If $S$ does not contain $e_2$, then it does not intersect $[u, v]$.

*Proof.* i) If $e_1 \in S$ and $e_2 \notin S$, then $S' = (S \setminus \{e_1\}) \cup \{e_2\}$ is in $S(G)$. As $ax^{S'} < \alpha$, we have a contradiction.
ii) Assume the contrary. Then we may suppose that $S$ contains an edge $e_i$, $i \in \{1, \ldots, p\} \setminus \{2\}$, and $e_2 \notin S$. Since $a(e_i) > a(e_2)$, this is impossible by the argument given above. ♦

Now, since $ax \geq \alpha$ is different from a trivial inequality, by Lemma 2.3.5 i), there is an edge set of $S_a(G)$, say $S_1$, containing $e_1$. Let $L_1$ be a 3-$st$-path of $S_1$ that contains $e_1$. By Claim 1 i), it follows that $e_2$ belongs to the second 3-$st$-path of $S_1$, say $L_2$. Note that $L_1 \cap L_2 = \emptyset$. It is not hard to see that $L_1$ and $L_2$ go through $e_1$ and $e_2$, respectively, in the same direction starting from $s$. If not, one would have one path of the form $(s, u, v, t)$ and the other one of the form $(s, v, u, t)$. But then, the edges $e_1, e_2$ might be deleted and one would obtain a feasible solution of weight smaller than $\alpha$, a contradiction. So, let us assume, w.l.o.g., that $u$ is the first node of $e_1, e_2$ used by $L_1, L_2$ going in this direction.

Let $L_1^s, L_1^t$ (resp. $L_2^s, L_2^t$) be the subpaths of $L_1$ (resp. $L_2$) between $s$ and $u$, and $v$ and $t$. Obviously, $|L_i^s \cup L_i^t| \leq 2$, for $i = 1, 2$. Note that we have either $L_1^s = \emptyset = L_2^s$ or $L_1^s \neq \emptyset \neq L_2^s$. Moreover, if the latter case holds, we have that $|L_1^t| \leq 1$ and $|L_2^t| \leq 1$. Note also that, by symmetry, these properties remain true if we exchange $s$ and $t$. Thus every $st$-path consisting of a combination of subpaths $L_i^s \cup \{e_j\} \cup L_k^t$ is of length at most 3, for $i, j, k = 1, 2$. In other words, we have that

$$|L_i^s \cup L_k^t| \leq 2, \text{ for all } i, k \in \{1, 2\}.$$

By Lemma 2.3.5 i), there must also exist an edge set of $S_a(G)$, say $S_2$, that does not contain $e_2$. By Claim 1 ii), we have that $[u,v] \cap S_2 = \emptyset$. Let $P_1$ and $P_2$ be two edge-disjoint 3-$st$-paths in $S_2$. We have the following claim.

*Claim 2.* At least one of the sets $P_1 \cap L_1$ and $P_2 \cap L_2$ ($P_2 \cap L_1$ and $P_1 \cap L_2$) is nonempty.

*Proof.* Assume, on the contrary, that for instance $P_1 \cap L_1 = \emptyset = P_2 \cap L_2$. Then, since $P_2 \cup L_2 \in S(G)$, it follows that $a(P_2) \geq a(L_1)$. Now, let $L_1' = (L_1 \setminus \{e_1\}) \cup \{e_2\}$. As $e_2 \notin S_2$ and hence $e_2 \notin P_1$, we have that $P_1 \cap L_1' = \emptyset$. Thus $P_1 \cup L_1' \in S(G)$ and therefore $a(L_1') \geq a(P_2)$. In consequence, $a(L_1') \geq a(L_1)$ and hence $a(e_2) \geq a(e_1)$, a contradiction. ♦

By Claim 2, we may assume, w.l.o.g., that $P_1 \cap L_2 \neq \emptyset$. Also by the same claim, at least one of the sets $P_1 \cap L_1$ and $P_2 \cap L_2$ is nonempty. In what follows, we suppose that $P_2 \cap L_2 \neq \emptyset$. The case where $P_1 \cap L_1 \neq \emptyset$ can be treated along the same line. As $e_2 \notin S_2$, it follows that $|L_2| = 3$. If $|L_2^s| = 2$, then $v = t$ and $L_2$ is of the form $(s, w, u, t)$ with $w \neq s, t, u$. Let $e_0$ be the edge of $L_2 \cap [u, w]$. Note that one of the 3-$st$-paths of $S_2$, say $P_1$, uses $e_0$. Then $P_1$ is of the form $(s, u, w, t)$. Let $\{f\} = P_1 \cap [w, t]$. As $(S_1 \setminus \{e_0, e_1\}) \cup \{f\}$ and $(S_2 \setminus \{e_0, f\}) \cup \{e_2\}$ are edge sets of $S(G)$, we obtain that $a(f) \geq a(e_0) + a(e_1)$ and $a(e_2) \geq a(e_0) + a(f)$, respectively. But this implies that $a(e_2) \geq a(e_1)$, a contradiction.

Consequently, $|L_2^s| \leq 1$ and, by symmetry, we also have that $|L_2^t| \leq 1$. Since $|L_2| = 3$, it follows that $|L_2^s| = |L_2^t| = 1$. So $L_1$ and $L_2$ are both of the form $(s, u, v, t)$. As $P_1 \cap L_2 \neq \emptyset \neq P_2 \cap L_2$ and $S_2 \cap [u, v] = \emptyset$, we may assume, w.l.o.g., that $P_1 \cap [s, u] \neq \emptyset$ and $P_2 \cap [v, t] \neq \emptyset$. Moreover, this implies that $P_1 \cap L_1 = \emptyset = P_2 \cap L_1$. Now, by replacing $e_1$ and $L_1^t$ by the subpath $P_1^{ut}$ of $P_1$ between $u$ and $t$, we get a solution, yielding $a(P_1^{ut}) \geq a(e_1) + a(L_1^t)$. Similarly, if we replace $P_1^{ut}$ by $e_2$ and $L_1^t$ in $S_2$, we obtain that $a(e_2) + a(L_1^t) \geq a(P_1^{ut})$. But this yields again $a(e_2) \geq a(e_1)$, which is impossible. □

By Lemma 2.3.7, multiple edges have the same coefficient in any non trivial facet-defining inequality of $P(G, L)$. For the rest of the paper, if $u, v \in N$, we will denote by $uv$ a fixed edge of $[u, v]$. If $P$ is a path of the form $(u_1, u_2, \ldots, u_q)$, then we will suppose that $P$ uses the edges $u_1u_2, \ldots, u_{q-1}u_q$. If for a solution $S \in S(G)$ and two nodes $u, v \in N$ we have that $S$ intersects $[u, v]$, then we will suppose that $S$ uses edge $uv$ and eventually further edges parallel to $uv$.

## 2.4 Structural properties

In this section we give some structural properties of the facet-defining inequalities of $P(G, L)$ different from the trivial and the $st$-cut inequalities. These will be useful for the proof of Theorem 2.3.1 in Section 2.5.

Let $L = 2, 3$ and $ax \geq \alpha$ be a facet-defining inequality of $P(G, L)$ different from the trivial and the $st$-cut inequalities. First, we give the following technical lemma, which will be frequently used in the subsequent proofs.

**Lemma 2.4.1.** *Let $S_1$ and $S_2$ be two edge sets of $S_a(G)$. Let $P_1$ and $P_1'$ be two edge-disjoint $L$-$st$-paths of $S_1$. Suppose that there is an $L$-$st$-path $P_2$ in $S_2$ such that $P_2 \cap P_1' = \emptyset$. Then, for every $L$-$st$-path $P$ not intersecting $S_2$, we have $a(P) \geq a(P_1)$.*

*Proof.* Let $S_1'$ (resp. $S_2'$) be the edge set obtained from $S_1$ (resp. $S_2$) by replacing $P_1$ by $P_2$ (resp. $P_2$ by $P$). As $S_1', S_2' \in S(G)$, it follows that $a(P_2) \geq a(P_1)$ and $a(P) \geq a(P_2)$. Hence, $a(P) \geq a(P_1)$. □

Now we show that the edges $e \in E$ such that $a(e) = 0$ do not form an $L$-$st$-path.

**Lemma 2.4.2.** *There cannot exist an $L$-$st$-path containing only edges with zero weight.*

*Proof.* We will show the result for $L = 3$. The proof for $L = 2$ can be done in a similar way.

Let us assume the contrary. Let $P_0$ be a shortest $st$-path such that $a(e) = 0$ for all $e \in P_0$. In what follows, we consider the case where $|P_0| = 3$. The cases where $|P_0| = 2$ or 1 can be treated similarly.

Let $P_0 = (s, u_1, u_2, t)$. Then $a(e) > 0$ for every chord of $P_0$. By Lemma 2.3.7, we have $a(e) = 0$ for all $e \in [s, u_1] \cup [u_1, u_2] \cup [u_2, t]$. As $ax \geq \alpha$ is different from a trivial inequality, by Lemma 2.3.5 i), there must exist an edge set $S$ of $S_a(G)$ not containing the edge $u_2t$ of $P_0$. Let $P_1, P_2$ be two edge-disjoint 3-$st$-paths of $S$.

*Claim 1.* Let $T$ be a solution of $S_a(G)$ and $T_1, T_2$ be two edge-disjoint 3-$st$-paths of $T$. Then at least one of the paths $T_1, T_2$ has only edges with zero value, if one of the following statements holds

i) $u_2t \notin T$,
ii) $su_1 \notin T$,
iii) $u_1u_2 \notin T$ and $|[u_2, t]| \geq 2$.

*Proof.* Suppose that both $T_1$ and $T_2$ use edges with positive weight. We first claim that both $T_1$ and $T_2$ intersect $P_0$. For otherwise, if for instance $T_1 \cap P_0 = \emptyset$, then $T_1 \cup P_0 \in S(G)$, yielding $a(P_0) \geq a(T_2)$. As $a(P_0) = 0$, we then have $a(T_2) = 0$, a contradiction.

Now suppose that $u_2t \notin T$. As $T_1 \cap T_2 = \emptyset$, one of the paths, say $T_1$, uses edge $u_1u_2$. Since $T_1$ uses at least one edge of positive weight, and $a(e) = 0$ for all $e \in [s, u_1] \cup [u_2, t]$, $T_1$ must be of the form $(s, u_2, u_1, t)$. By the remark above, we have indeed that $a(u_1t) > 0$. Now if we replace in $T$ the edges $u_1u_2$ and $u_1t$ by $u_2t$, we get a solution of $S(G)$. Moreover, as $a(u_2t) = a(u_1u_2) = 0$, it follows that $a(u_1t) = 0$, a contradiction.

If $su_1 \notin T$, then the statement follows by symmetry.

Suppose now that $u_1u_2 \notin T$ and $|[u_2, t]| \geq 2$. Denote by $f$ an edge of $[u_2, t] \backslash \{u_2t\}$. Since $u_1u_2 \notin T$ and $T_1 \cap P_0 \neq \emptyset \neq T_2 \cap P_0$, we may suppose, w.l.o.g., that $su_1 \in T_1$ and $u_2t \in T_2$. Let $T_1^{u_1t}$ be the subpath of $T_1$ between $u_1$ and $t$. Observe that $a(T_1^{u_1t}) > 0$. Consider the solution obtained from $T$ by replacing $T_1^{u_1t}$ by the edges $u_1u_2$ and $f$. As $a(f) = a(u_1u_2) = 0$, this yields $a(T_1^{u_1t}) = 0$, a contradiction, which ends the proof of the claim. ♦

As $u_2t \notin S$, by Claim 1 i), it follows that at least one of the paths $P_1$ and $P_2$, say $P_1$, contains only edges with zero coefficient. Moreover, we have that $P_1 \cap P_0 \neq \emptyset$. For otherwise, there would exist a solution formed by $P_1$ and $P_0$ of weight zero, contradicting the fact that $\alpha > 0$.

*Claim 2.* i) $|[u_2, t]| \geq 2$.
ii) $|[s, u_1]| \geq 2$.

*Proof.* We will prove i), the proof of ii) follows by symmetry. Suppose that $|[u_2, t]| = 1$. We claim that the edge $su_1$ of $P_0$ belongs to $P_1$. In fact, if this is not the case, as $u_2t \notin S$ and $P_1 \cap P_0 \neq \emptyset$, $P_1$ must contain the edge $u_1u_2$. As $|[u_2, t]| = 1$ and $u_2t \notin S$, $P_1$ must use an edge of $[u_1, t]$ which is of positive weight, a contradiction. Thus $P_1$ is of the form $(s, u_1, v, t)$ with $v \neq u_2$. We thus have $|[s, u_1]| = 1$. For otherwise, we would have two edge-disjoint 3-$st$-paths of zero weight, yielding $\alpha = 0$, a contradiction. By considering a solution of $S_a(G)$ not containing $su_1$, and using Claim 1 ii) together with similar arguments as above, we can show that there exists a path $P_1'$ of the form $(s, w, u_2, t)$, with $w \neq u_1$, constituted of edges with zero coefficient. As $P_1$ and $P_1'$ are edge-disjoint and hence form a solution of $S(G)$, this yields $\alpha = 0$, a contradiction. ♦

Since there are not two edge-disjoint 3-$st$-paths of weight zero, at least one of the sets $[s, u_1]$, $[u_1, u_2]$, $[u_2, t]$ must be reduced to a single edge. Consequently, by Claim 2, it follows that $|[u_1, u_2]| = 1$. Consider now a solution $S'$ of $S_a(G)$ not containing $u_1u_2$. Let $P_1'$ and $P_2'$ be two edge-disjoint 3-$st$-paths of $S'$. As by Claim 2 ii), $|[u_2, t]| \geq 2$, we may, w.l.o.g., suppose by Claim 1 iii) that $a(P_1') = 0$. Also since $\alpha > 0$, one should have $P_1' \cap P_0 \neq \emptyset$. Since $u_1u_2 \notin S'$, we may, w.l.o.g., suppose that $su_1 \in P_1'$. Therefore $P_1' = (s, u_1, v', t)$ with $v' \neq u_2$. As, by Claim 2, $|[s, u_1]| \geq 2$, the solution given by $P_0 \cup \tilde{P}_1$ where $\tilde{P}_1 = (f, u_1v', v't)$ with $f \in [s, u_1] \backslash \{su_1\}$ would be in $S(G)$ and of zero weight. But this is a contradiction, and the proof of the lemma is complete. □

Let us denote by $U$ (resp. $V$) the subset of nodes $u$ such that $a(e) = 0$ for all $e \in [s, u]$ (resp. $e \in [u, t]$). Note that, by Lemma 2.3.7, if for an edge $f \in [s, u]$ (resp. $f \in [u, t]$) for some $u \in N \backslash \{s, t\}$, we have $a(f) = 0$, then $u \in U$ (resp. $u \in V$). By Lemma 2.4.2, we have that $U \cap V = \emptyset$. Moreover, $a(e) > 0$ for all $e \in [s, t] \cup [s, V] \cup [U, t]$. If $L = 3$, we also have that $a(e) > 0$ for all $e \in [U, V]$. Let $W = N \backslash (\{s, t\} \cup U \cup V)$. Note that, if $W \neq \emptyset$, $a(e) > 0$ for all $e \in [s, W] \cup [W, t]$.

**Lemma 2.4.3.** *$U \neq \emptyset \neq V$.*

*Proof.* We will prove the lemma for $U$. The proof for $V$ is similar. Since $ax \geq \alpha$ is different from the $st$-cut constraint corresponding to the node $s$, by Lemma 2.3.5 ii), there is an edge set $F$ of $S_a(G)$ that contains at least three edges of $\delta(s)$. As only two of these edges can be used by two edge-disjoint $L$-$st$-paths of $F$, there is an edge of $F \cap \delta(s)$, say $e_0 \in [s, u]$ with $u \in N \backslash \{s, t\}$, such that $F \backslash \{e_0\} \in S(G)$. This implies that $a(e_0) = 0$, and therefore $u \in U$. □

This technical lemma will be used in the subsequent proofs.

**Lemma 2.4.4.** *Let $S \in S_a(G)$ and $P_1$ be a 3-$st$-path of $S$ going through a node $u$ of $N \backslash \{s, t\}$. Let $\tilde{P}_1$ be the subpath of $P_1$ between $s$ (resp. $t$) and $u$. Let $P$ be a path between $s$ (resp. $t$) and $u$ such that $a(P) = 0$ and $|P| \leq |\tilde{P}_1|$. If $a(\tilde{P}_1) > 0$, then $P \cap P_2 \neq \emptyset$ for any 3-$st$-path $P_2$ of $S$, where $P_2 \cap P_1 = \emptyset$.*

*Proof.* If $P \cap P_2 = \emptyset$, as $|P| \leq |\tilde{P}_1|$, the edge set $(S \setminus \tilde{P}_1) \cup P$ belongs to $S(G)$ and hence $a(\tilde{P}_1) \leq a(P)$. As $a(P) = 0$ and $a(\tilde{P}_1) > 0$, this is impossible. □

The following lemma shows that the edges having both end nodes in $U$ $(V)$ all have zero coefficient. Moreover, if $L = 2$, the same holds for the edges between $U$ and $V$.

**Lemma 2.4.5.** *i) If $L = 2$, then $a(e) = 0$ for all $e \in [U, V]$.*
*ii) $a(e) = 0$, for all $e \in E(U) \cup E(V)$.*

*Proof.* i) Let $e \in [U, V]$ and let $S$ be a solution of $S_a(G)$ containing $e$. As $e$ cannot belong to a 2-$st$-path of $S$, $S \setminus \{e\}$ is also a solution of $S(G)$, and therefore $a(e) = 0$.

ii) If $L = 2$ and $e \in E(U) \cup E(V)$, we can show as in i) that $a(e) = 0$. Now let us consider the case where $L = 3$. Let us assume, on the contrary, that there exists an edge $u_1u_2$ with $u_1, u_2 \in U$ (the case where $u_1, u_2 \in V$ is similar) such that $a(u_1u_2) > 0$. Note that by Lemma 2.3.7 it follows that $a(e) > 0$ for all $e \in [u_1, u_2]$. Let us consider an edge set of $S_a(G)$, say $S_1$, that contains $u_1u_2$ and let $P_1, P'_1$ be two edge-disjoint 3-$st$-paths in $S_1$. As $a(u_1u_2) > 0$, $u_1u_2$ must be in one of the 3-$st$-paths of $S_1$, say $P_1$. We can suppose, w.l.o.g., that $P_1$ is $(s, u_1, u_2, t)$. Moreover, as $a(e) = 0$ for all $e \in [s, u_2]$, by Lemma 2.4.4, $P'_1$ must contain every edge of $[s, u_2]$. However, this is possible only if $|[s, u_2]| = 1$. Consequently, we will assume in the rest of the proof that $[s, u_2] = \{su_2\}$ and $su_2 \in P'_1$. Let us assume that $P'_1$ is of the form $(s, u_2, z, t)$ with $z \neq s, t, u_2$. If $P'_1$ consists of only two edges, then the proof is similar. Furthermore, $z \notin U$. For otherwise, one can consider the edge set $S'_1 = (S_1 \setminus \{su_1, u_1u_2, u_2z\}) \cup \{sz\}$, which is a solution of $S(G)$. As $a(sz) = 0$, we get $a(su_1) + a(u_1u_2) + a(u_2z) \leq 0$ and hence $a(u_1u_2) = 0$, a contradiction. Therefore $z \in V \cup W$.

Moreover, we have that $a(e) > 0$ for all $e \in [U \setminus \{u_1, u_2\}, u_2]$. Indeed, if $a(e) = 0$, then the edge set $(S_1 \setminus \{su_1, u_1u_2\}) \cup \{su, e\}$, where $u$ is the end node of $e$ different from $u_2$, would be a solution of $S(G)$ with a weight smaller than $\alpha$, a contradiction.

Now, let us consider an edge set of $S_a(G)$, say $S_2$, that does not contain the edge $su_2$. Let $P_2, P'_2$ be two edge-disjoint 3-$st$-paths in $S_2$. We claim that $[u_2, t] \cap S_2 = \emptyset$. In fact, if one of the 3-$st$-paths of $S_2$, say $P_2$, uses an edge of $[u_2, t]$, say $u_2t$, as $|[s, u_2]| = 1$ and $su_2 \notin S_2$, one should have $P_2 = (sw, wu_2, u_2t)$ where $w \in N \setminus \{s, u_2, t\}$. Moreover, we have $a(sw) + a(wu_2) > 0$. In fact, this is clear if $w \notin U$. If $w \in U$, then, as shown above, $a(wu_2) > 0$ and the statement follows. Now, by replacing in $S_2$ the subpath $(sw, wu_2)$ by $su_2$, we get a solution of smaller weight, which is impossible.

Thus $[u_2, t] \cap S_2 = \emptyset$ and hence, as $su_2 \notin S_2$, no 3-$st$-path in $S_2$ goes through the node $u_2$. Let $P$ be the path $(su_2, u_2t)$. Thus, $P \cap S_2 = \emptyset$. Moreover, as neither $su_2$ nor $u_2z$ belongs to $S_2$, at most one of the paths $P_2, P_2'$ intersects $P_1'$. W.l.o.g., we may suppose that $P_2 \cap P_1' = \emptyset$. From Lemma 2.4.1, it then follows that $a(P) \geq a(P_1)$. But this implies that $a(u_1u_2) = 0$, a contradiction. □

Now we show that the subset $W$ is empty when $L = 2$, and nonempty when $L = 3$.

**Lemma 2.4.6.** *i) If $L = 2$, then $W = \emptyset$.*
*ii) If $L = 3$, then $W \neq \emptyset$.*

*Proof.* i) Assume the contrary, and let $w \in W$. Then $a(e) > 0$ for all $e \in [s, w] \cup [w, t]$. We will show that $|[s, w] \cap F| = |[w, t] \cap F|$ for every $F \in S_a(G)$. In fact, suppose, by contradiction, that there exists $F \in S_a(G)$ such that, for instance, $|[s, w] \cap F| > |[w, t] \cap F|$. Since at most $|[w, t] \cap F|$ edge-disjoint 2-$st$-paths can go through $w$, there must exist an edge, say $\bar{e}$, of $[s, w] \cap F$ such that $F \backslash \{\bar{e}\} \in S(G)$. This implies that $a(\bar{e}) = 0$, a contradiction. Thus, the incidence vector of any solution of $S_a(G)$ verifies the equation $x([s, w]) = x([w, t])$. As, by Lemma 2.3.6, this equation cannot be a positive multiple of $ax \geq \alpha$, we get a contradiction.

ii) Assume that, on the contrary, $W = \emptyset$. Let $U' = U \cup \{s\}$. Since $ax \geq \alpha$ is different from the $st$-cut inequality associated to $\delta(U')$, there exists an edge set of $S_a(G)$, say $F_1$, that uses at least three edges of $\delta(U')$. Let $P_1, P_1'$ be two edge-disjoint 3-$st$-paths of $F_1$. Since $W = \emptyset$, $a(e) > 0$ for all $e \in \delta(U')$, and hence every edge of $F_1 \cap \delta(U')$ must belong to one of the paths $P_1$ and $P_1'$. So, one of these paths, say $P_1$, must use at least two edges of $\delta(U')$. As any $st$-path intersects any $st$-cut an odd number of times, we have that $P_1$ contains exactly three edges of $\delta(U')$. Therefore, $P_1$ is of the form $(s, v, u, t)$, where $u \in U$ and $v \in V$. Let $F_2 = (F_1 \backslash (P_1' \cup \{vu\})) \cup \{su, vt\}$. Obviously, $F_2 \in S(G)$. As $a(su) = a(vt) = 0$, it follows that $a(vu) = 0$, a contradiction. □

For the rest of this section, we assume that $L = 3$.

**Lemma 2.4.7.** *i) If there are a node $w \in W$ and a node $u_1 \in U$ such that $a(u_1w) = 0$, then $a(e) = 0$ for all $e \in [U, w]$.*
*ii) If there are a node $w \in W$ and a node $v_1 \in V$ such that $a(wv_1) = 0$, then $a(e) = 0$ for all $e \in [w, V]$.*

*Proof.* We show the result for $U$, the proof for $V$ is similar. If $|U| = 1$, then the statement follows from Lemma 2.3.7. So, let us suppose that $|U| \geq 2$ and assume, on the contrary, that there is a node $u_2 \in U$ such that $a(u_2w) > 0$. Let $S_1$ be a solution of $S_a(G)$ such that $u_2w \in S_1$. As $a(u_2w) > 0$, $u_2w$ must belong to a 3-$st$-path $P_1$ in $S_1$. Let $P_1'$ be a further 3-$st$-path of $S_1$ with $P_1 \cap P_1' = \emptyset$.

*Claim 1.* $P_1 = (s, u_2, w, t)$.

*Proof.* As $u_2w \in P_1$, $P_1$ is of the form either $(s, w, u_2, t)$ or $(s, u_2, w, t)$. Suppose that the first case holds. As $a(e) > 0$ for all $e \in [s, w]$ and $a(e) = 0$ for all $e \in [s, u_2]$, it follows from Lemma 2.4.4 that $P_1'$ uses all the edges between $s$ and $u_2$. Therefore $[s, u_2] \subseteq P_1'$. Moreover, since, by Lemma 2.4.5 ii), all the 2-$su_2$-paths going through $u_1$ have weight zero, again by Lemma 2.4.4, $P_1'$ must also intersect all these paths. As $P_1'$ cannot use more than one edge incident to $s$, one should have $[u_1, u_2] \subseteq P_1'$. In consequence, $|[s, u_2]| = |[u_1, u_2]| = 1$ and $P_1'$ is of the form $(s, u_2, u_1, t)$. But, by adding edge $su_1$ and removing the edges $sw, wu_2$, we obtain a solution of lower weight, which is impossible.♦

Consequently, $P_1 = (s, u_2, w, t)$. As $a(u_2w) > 0$, and therefore the weight of the subpath of $P_1$ between $s$ and $w$ is positive, it follows by Lemma 2.4.4 that $P_1'$ must intersect every 2-$sw$-path of weight zero going through $u_1$. Since $a(u_1w) = 0$, by Lemma 2.3.7, $a(e) = 0$ for all $e \in [u_1, w]$. Thus, as $a(e) = 0$ for all $e \in [s, u_1]$, we obtain that at least one of the sets $[s, u_1]$ and $[u_1, w]$ is reduced to a single edge. If there is a node $u \in U \backslash \{u_1, u_2\}$ such that $a(e) = 0$ for some edge $e \in [u, w]$, then by Lemma 2.4.4, $P_1'$ must also intersect the 2-$sw$-paths going through $u$. But as $|P_1'| \leq 3$, this is not possible. Therefore $a(e) > 0$ for all $e \in [U \backslash \{u_1\}, w]$.

*Claim 2.* $P_1' \cap [u_1, w] = \emptyset$.

*Proof.* Suppose, on the contrary, that $P_1'$ uses for instance $u_1w$. If $P_1' = (s, w, u_1, t)$, then, as the weight of the subpath of $P_1'$ between $s$ and $u_1$ is positive, and $a(e) = 0$ for all $e \in [s, u_1]$, by Lemma 2.4.4 it follows that $P_1$ uses all the edges between $s$ and $u_1$. But this contradicts Claim 1. Hence $P_1'$ is of the form $(su_1, u_1w, h)$ where $h \in [w, t] \backslash \{wt\}$. We consider two cases.

**Case 1.** $|[s, u_1]| = 1$. Consider an edge set $S_2$ of $S_a(G)$ such that $su_1 \notin S_2$. We may suppose that $S_2$ is minimal. Let $P_2$ and $P_2'$ be the two edge-disjoint 3-$st$-paths of

$S_2$. If $S_2$ uses an edge $u_1z$ with $z \in V \cup W$, then $u_1z$ belongs to one of the 3-$st$-paths of $S_2$, say $P_2$. As $su_1 \notin S_2$, $P_2 = (s, z, u_1, t)$. Observe that $a(e) > 0$ for all $e \in [s, z]$. Now by replacing the edges $sz, zu_1$ by $su_1$, we get a solution of $S(G)$ of weight less than $\alpha$, a contradiction. In consequence, we have $[u_1, V \cup W] \cap S_2 = \emptyset$ and therefore $[u_1, w] \cap S_2 = \emptyset$. Suppose now that $S_2 \cap [w, t] \neq \emptyset$ and for instance that $P_2 \cap [w, t] \neq \emptyset$. Since $a(e) > 0$ for all $e \in [U \backslash \{u_1\}, w]$, the subpath of $P_2$, say $P_2^{sw}$, between $s$ and $w$ has a positive weight. As $\{su_1, u_1w\}$ is a 2-$sw$-path of weight zero which does not intersect $S_2$, if we replace $P_2^{sw}$ by $su_1, u_1w$, we get a solution of lower weight, which is impossible. Thus $S_2 \cap [w, t] = \emptyset$, and, in consequence, $P_1' \cap S_2 = \emptyset$. Let $P = P_1'$. By Lemma 2.4.1, it follows that $a(P) = a(P_1') \geq a(P_1)$. As $a(h) = a(wt)$ and $a(su_1) = a(u_1w) = 0$, this yields $a(u_2w) = 0$, a contradiction.

**Case 2.** $|[s, u_1]| \geq 2$. Since one of the sets $[s, u_1], [u_1w]$ contains exactly one edge, we have that $[u_1, w] = \{u_1w\}$. Let $\bar{S}_2$ be a solution of $S_a(G)$ not containing $u_1w$. Suppose that $\bar{S}_2$ is minimal, and let $\bar{P}_2$ and $\bar{P}_2'$ be the two edge-disjoint 3-$st$-paths of $\bar{S}_2$. We can show, in a similar way as in Case 1, that $[w, t] \cap \bar{S}_2 = \emptyset$. As $u_1w \notin \bar{S}_2$, it follows that $|\bar{S}_2 \cap P_1'| \leq 1$. Hence, there is a 3-$st$-path of $\bar{S}_2$, say $\bar{P}_2$, that does not intersect $P_1'$. Therefore $\bar{P}_2 \cup P_1'$ is a solution of $S(G)$, yielding $a(\bar{P}_2) \geq a(P_1)$. On the other hand, since $|[s, u_1]| \geq 2$, we may suppose that $\bar{P}_2' \cap P_1' = \emptyset$. So, if we replace, in $\bar{S}_2$, $\bar{P}_2$ by $P_1'$, we get a solution of $S(G)$, implying that $a(P_1') \geq a(\bar{P}_2)$. Therefore $a(P_1') \geq a(P_1)$ and hence $a(u_2w) = 0$, a contradiction. ♦

By Claim 2, we then have $P_1' \cap [u_1, w] = \emptyset$. As $P_1'$ intersects all the 2-$sw$-paths going through $u_1$, it follows that $[s, u_1] = \{su_1\}$ and $su_1 \in P_1'$.

If $P_1'$ uses an edge of $[u_1, t]$, then, by removing the edge $u_2w$ and adding edges $u_1w$ and $u_1u_2$, we get a solution of $S(G)$. But this implies that $a(u_2w) = 0$, which is impossible. Along the same line, we can also show that $P_1'$ does not go through any node of $U$. Hence $P_1'$ must use a node of $V \cup W$, say $v$.

Consider now a solution $S_3$ of $S_a(G)$ not containing $su_1$. Let $P_3$ and $P_3'$ be two edge-disjoint 3-$st$-paths of $S_3$. Suppose that there is an edge, say $u_1z$, of $[u_1, V \cup W]$ that belongs to $S_3$. Since $su_1 \notin S_3$, the 3-$st$-path containing $u_1z$, say $P_3$, must be of the form $(s, z, u_1, t)$. Note that the subpath between $s$ and $u_1$ has a positive weight. As $a(su_1) = 0$, by Lemma 2.4.4, it follows that $su_1 \in P_3'$ and hence $su_1 \in S_3$, contradicting our hypothesis. Thus $[u_1, V \cup W] \cap S_3 = \emptyset$ and hence $([u_1, v] \cup [u_1, w]) \cap S_3 = \emptyset$. Thus $|P_1' \cap S_3| \leq 1$. Consequently, there must exist a 3-$st$-path of $S_3$, say $P_3$, such that $P_1' \cap P_3 = \emptyset$. Also we may show in a similar way that $[w, t] \cap S_3 = \emptyset$. Consider now the path $P = (s, u_1, w, t)$. Observe that $P \cap S_3 = \emptyset$. By Lemma 2.4.1 with respect to $S_1$ and $S_3$, it follows that $a(P) \geq a(P_1)$. But this implies that $a(u_2w) = 0$, a contradiction, and the proof of the lemma is complete. $\square$

We now establish the equality of coefficients $a(e)$ for all edges $e$ between $U$ and $t$, or between $s$ and $V$.

**Lemma 2.4.8.** *For all $e, e' \in [U,t]$ (resp. $e, e' \in [s,V]$), $a(e) = a(e')$.*

*Proof.* We will prove the lemma for $U$, the proof for $V$ is similar. If $|U| = 1$, the statement follows from Lemma 2.3.7. So suppose $|U| \geq 2$. Let $u_1, u_2 \in U$ such that $a(u_1t) = \min\{a(e), e \in [U,t]\}$ and $a(u_2t) = \max\{a(e), e \in [U,t]\}$. Assume that $a(u_2t) > a(u_1t)$.

*Claim.* i) Let $S \in S_a(G)$. If $S \cap [u_2,t] \neq \emptyset$, then $[u_1,t] \subseteq S$.
ii) $|[u_1,t]| = 1$.

*Proof.* i) Suppose that $u_2t \in S$, and let $T_1$ and $T_2$ be two edge-disjoint 3-$st$-paths of $S$. As $a(u_2t) > 0$, we may suppose, for instance, that $u_2t \in T_2$. Assume that there is an edge $e_1$ of $[u_1,t]$ that is not in $S$. If there is an edge $e \in [s,u_1]$ that is not in $T_1$, then we can replace $u_2t$ by $e$ and $e_1$ and get a solution of $S(G)$ of lower weight, a contradiction. Hence $[s,u_1] \subseteq T_1$ and therefore $[s,u_1] = \{su_1\}$, $su_1 \in T_1$, and $[s,u_2] \cap T_1 = \emptyset$. Furthermore, if $T_1$ contains an edge $e' \in [u_1,u_2]$, then, as $su_1 \in T_1$, $T_1$ must use an edge $f$ of $[u_2,t] \backslash \{u_2t\}$. Now it is easy to see that $(S \backslash \{f\}) \cup \{e_1\} \in S(G)$. Since by Lemma 2.3.7, $a(e_1) = a(u_1t)$ and $a(f) = a(u_2t)$, it follows that $a(u_1t) \geq a(u_2t)$. But this contradicts our hypothesis. Therefore $[u_1,u_2] \cap T_1 = \emptyset$. Consider now the solution $S' = (S \backslash \{u_2t\}) \cup \{su_2, u_1u_2, e_1\}$. As $a(su_2) = a(u_1u_2) = 0$, we have that $a(u_1t) = a(e_1) \geq a(u_2t)$, a contradiction.
ii) Let $\bar{S} \in S_a(G)$ such that $u_2t \in \bar{S}$. We may suppose $\bar{S}$ minimal. Let $\overline{T}_1, \overline{T}_2$ be the edge-disjoint 3-$st$-paths of $\bar{S}$, and suppose, w.l.o.g., that $u_2t \in \overline{T}_2$. From i), it follows that $[u_1,t] \subseteq \bar{S}$. Moreover, as $u_2t \in \overline{T}_2$, $\overline{T}_2 \cap [u_1,t] = \emptyset$, and hence $[u_1,t] \subseteq \overline{T}_1$. This implies that $|[u_1,t]| = 1$. $\blacklozenge$

Let $S_1$ be a solution of $S_a(G)$ containing $u_2t$. By the claim above, $S_1$ also contains $u_1t$. As $a(su_1) = a(su_2) = 0$, and $\{su_1, su_2, u_1t, u_2t\}$ is a solution of $S(G)$, we may assume that $S_1 = \{su_1, su_2, u_1t, u_2t\}$.

Consider now a solution $S_2 \in S_a(G)$ that does not contain $u_1t$, which may be supposed minimal. Since $u_1t \notin S_2$, by the claim it follows that $[u_2,t] \cap S_2 = \emptyset$. And, in consequence, $[u_1, u_2] \cap S_2 = \emptyset$. Suppose that $S_2$ contains an edge $su_1$. Since $S_2$ is minimal, one of the two 3-$st$-paths of $S_2$, say $T$, contains $su_1$, and hence $T$ is of the form $(s, u_1, z, t)$ where $z \in N \backslash \{s, t, u_1, u_2\}$. Let $T^{u_1t}$ be the subpath of $T$ between $u_1$ and $t$. As the sets $(S_2 \backslash T^{u_1t}) \cup \{u_1t\}$ and $(S_1 \backslash \{u_2t\}) \cup (\{u_1u_2\} \cup T^{u_1t})$ are both solutions of $S(G)$, and, as by Lemma 2.4.5 ii) $a(u_1u_2) = 0$, we have that $a(u_1t) \geq a(T^{u_1t}) \geq a(u_2t)$, a contradiction. Consequently, $[s, u_1] \cap S_2 = \emptyset$.

Let $P_1 = (su_2, u_2t)$ and $P_1' = (su_1, u_1t)$ be the two 3-$st$-paths of $S_1$. Let $P = P_1'$ and $P_2$ be any 3-$st$-path of $S_2$. Note that $P \cap S_2 = P_1' \cap S_2 = \emptyset$, and hence $P_2 \cap P_1' = \emptyset$. By Lemma 2.4.1, it follows that $a(P) \geq a(P_1)$. However, as $a(su_1) = a(su_2) = 0$, this implies again that $a(u_1t) \geq a(u_2t)$, which is impossible. □

Here under, we prove that, if $U = \{u\}$ (resp. $V = \{v\}$), any minimal solution in $S_a(G)$ not using edges in $[s, u]$ (resp. $[v, t]$) does not go through node $u$ (resp. $v$) at all.

**Lemma 2.4.9.** *Let $S$ be a minimal solution of $S_a(G)$.*
*i) If $U = \{u\}$ and $S \cap [s, u] = \emptyset$, then $\delta(u) \cap S = \emptyset$.*
*ii) If $V = \{v\}$ and $S \cap [v, t] = \emptyset$, then $\delta(v) \cap S = \emptyset$.*

*Proof.* We will show i), the proof of ii) is similar. We first show that $[u, t] \cap S = \emptyset$. Assume, on the contrary, that $ut \in S$. Then, as $a(ut) > 0$, one of the 3-$st$-paths of $S$, say $P$, must contain $ut$. As $[s, u] \cap S = \emptyset$, $P$ must be of the form $(s, w, u, t)$, where $w \in N \backslash \{s, t, u\}$. Note that $w \notin U$, and hence $a(sw) > 0$. Thus, one can replace $sw$ and $wu$ by $su$ in $S$ and get a solution of $S(G)$ of weight less than $\alpha$, a contradiction. Thus $[u, t] \cap S = \emptyset$. Now, by the minimality of $S$, no other edge of $\delta(u)$ may be used by $S$. □

The following lemma shows that all the edges in $[U, t] \cup [s, V]$ have the same coefficient in $a$.

**Lemma 2.4.10.** *$a(e) = a(e')$ for all $e \in [U, t]$ and $e' \in [s, V]$.*

*Proof.* Assume the contrary. Thus, by Lemma 2.4.8, we may assume, w.l.o.g., that

$$a(e) > a(e') \text{ for all } e \in [U, t] \text{ and } e' \in [s, V]. \tag{2.4}$$

Let $u_1 \in U$. Consider a solution $S_1$ of $S_a(G)$ that contains $u_1t$ and suppose that $S_1$ is minimal. Let $P_1$ and $P_1'$ be the two edge-disjoint 3-$st$-paths of $S_1$, and suppose that

$u_1t \in P_1$.

*Claim.* $|V| = 1$.

*Proof.* Assume that $|V| \geq 2$. First observe that $P_1$ cannot go through a node $v \in V$. For otherwise, $P_1$ would be of the form $(s, v, u_1, t)$. Since the subpaths of $P_1$ between $s$ and $u_1$, and $v$ and $t$, have both positive weight, by Lemma 2.4.4, $P_1'$ must use edges $su_1$ and $vt$. Now, if we remove the edges of $S_1$ between $u_1$ and $v$, we still have a solution of $S(G)$. This implies that $a([u_1, v]) = 0$. But this contradicts the fact that $a(u_1v) > 0$. In consequence, since $S_1$ is minimal, $S_1$ may contain at most one edge from $[s, V]$. Suppose that $S_1$ contains edge $sv_1$, where $v_1 \in V$. Note that $sv_1 \in P_1'$. As $|V| \geq 2$, there is an edge $sv_2$, with $v_2 \in V$, that does not belong to $S_1$. If there is an edge $e \in [v_2, t]$ such that $e \notin S_1$, then, by replacing $u_1t$ by $sv_2$ and $e$, we get a solution of $S(G)$. As $a(e) = 0$, this yields $a(sv_2) \geq a(u_1t)$, which contradicts (2.4). Thus $[v_2, t] \subseteq S_1$ and therefore $[v_2, t] \subseteq P_1'$. This implies that $[v_2, t] = \{v_2t\}$ and $P_1' = (s, v_1, v_2, t)$. By considering the solution obtained by replacing $u_1t$ by $sv_2$ and $v_1t$, we obtain that $a(sv_2) \geq a(u_1t)$, which once again contradicts (2.4).

Consequently, $S_1 \cap [s, V] = \emptyset$. Now remark that, since $S_1$ is minimal and $u_1t \in S_1$, $S_1$ cannot use two edges of $[V, t]$. Thus there is a node $z \in V$ such that $([s, z] \cup [z, t]) \cap S_1 = \emptyset$. By replacing $u_1t$ by $sz$ and $zt$ in $S_1$, we get a solution of $S(G)$, yielding $a(sz) \geq a(u_1t)$. This contradicts (2.4), and the claim is proved. ♦

Let $V = \{v\}$. Let $P = (s, v, t)$ be an $st$-path of length 2 going through $v$. We claim that $P_1' \cap P \neq \emptyset$. In fact, if this is not the case, then, as the edge set obtained from $S_1$ by replacing $P_1$ by $P$ is in $S(G)$, we would have that $a(sv) \geq a(u_1t)$. But this contradicts (2.4). Therefore, $P_1'$ must contain at least one of the sets $[s, v]$ and $[v, t]$. Thus at least one of the sets $[s, v]$ and $[v, t]$ is reduced to a single edge.

**Case 1.** $[v, t] = \{vt\}$. Consider a solution $S_2 \in S_a(G)$ not containing $vt$, which is supposed minimal. Then, by Lemma 2.4.9, $S_2 \cap \delta(v) = \emptyset$ and hence $P \cap S_2 = \emptyset$. Moreover, as $P_1' \cap P \neq \emptyset$, $P_1'$ does meet $v$, and therefore $|P_1' \cap S_2| \leq 1$. Thus there exists a 3-$st$-path of $S_2$, say $P_2$, that does not intersect $P_1'$. As $P \cap S_2 = \emptyset$, by Lemma 2.4.1, we have that $a(P) \geq a(P_1)$, and hence $a(sv) \geq a(u_1t)$. But this contradicts (2.4).

**Case 2.** $[s, v] = \{sv\}$. By Case 1, we may suppose that $|[v, t]| \geq 2$. As $P_1'$ contains one of the sets $[s, v]$ and $[v, t]$, it follows that $sv \in P_1'$. Note that $\{su_1, u_1t, sv, vt\} \in$

$S(G)$. As $a(su_1) = a(vt) = 0$ and $S_1$ is minimal, we may suppose, w.l.o.g., that $S_1 = \{su_1, u_1t, sv, vt\}$. Hence $P_1 = (su_1, u_1t)$ and $P_1' = (sv, vt)$. Consider now an edge set $S_3$ of $S_a(G)$ not containing $sv$ and suppose that $S_3$ is minimal. Since $|P_1' \cap S_3| \leq 1$, there must exist a 3-$st$-path in $S_3$, say $P_3$, such that $P_3 \cap P_1' = \emptyset$. If we replace, in $S_1$, $P_1$ by $P_3$, the resulting set is still a solution of $S(G)$, and therefore $a(P_3) \geq a(P_1)$. On the other hand, if there is an edge $h \in [v,t]$ such that $h \notin S_3$, then one can replace the path $P_3$ by the one formed by $sv$ and $h$, and get a solution of $S(G)$. But this implies that $a(P_3) \leq a(sv) + a(h)$. As $a(P_3) \geq a(P_1)$ and $a(h) = 0$, we obtain that $a(u_1t) \leq a(sv)$, contradicting (2.4). Thus $[v,t] \subseteq S_3$. As $|[v,t]| \geq 2$ and $S_3$ is minimal, it follows that $P_3 \cap [v,t] \neq \emptyset$. Let $P_3^{sv}$ be the subpath of $P_3$ between $s$ and $v$. By replacing, in $S_3$, $P_3^{sv}$ by $sv$, we get a solution of $S(G)$, which yields $a(sv) \geq a(P_3^{sv})$. As $a(P_3) \geq a(P_1)$, and therefore $a(P_3^{sv}) \geq a(u_1t)$, we get $a(sv) \geq a(u_1t)$. But this contradicts again (2.4), which ends the proof of the lemma. □

Lemma 2.4.7 allows a partition of the set $W$ into four subsets:
$W_1 = \{w \in W \mid a(e) = 0 \text{ for all } e \in [U,w], \text{ and } a(e') > 0 \text{ for all } e' \in [w,V]\}$,
$W_2 = \{w \in W \mid a(e) = 0 \text{ for all } e \in [U,w] \cup [w,V]\}$,
$W_3 = \{w \in W \mid a(e) > 0 \text{ for all } e \in [U,w], \text{ and } a(e') = 0 \text{ for all } e' \in [w,V]\}$,
$Z = W \backslash (W_1 \cup W_2 \cup W_3)$.

**Lemma 2.4.11.** *i) If $U = \{u\}$, then $a(e) = a(e')$, for all $e \in [u,t]$ and $e' \in [W_1 \cup W_2, t]$. ii) If $V = \{v\}$, then $a(e) = a(e')$, for all $e \in [s,v]$ and $e' \in [s, W_2 \cup W_3]$.*

*Proof.* We will only prove i), the proof of ii) is similar. Assume by contradiction that $a(ut) \neq a(wt)$ for some $w \in W_1 \cup W_2$. We will first give the following claim.

*Claim.* No solution of $S_a(G)$ uses at the same time an edge of $[u,t]$ and an edge of $[w,t]$.

*Proof.* It suffices to show that there is no solution of $S_a(G)$ containing at the same time $ut$ and $wt$. Let us suppose, on the contrary, that there exists a solution $S \in S_a(G)$ with $ut, wt \in S$. Let $T_1$ and $T_2$ be two edge-disjoint 3-$st$-paths of $S$. As $a(ut) > 0$ and $a(wt) > 0$, we may suppose that $ut \in T_1$ and $wt \in T_2$.

Suppose that $a(wt) < a(ut)$. The case where $a(wt) > a(ut)$ can be treated along the same line. If $[s,u] \cap T_1 = \emptyset$, $T_1$ must go through a node $z \in N \backslash \{s,t,u\}$, and hence the subpath $T_1^{su}$ of $T_1$ between $s$ and $u$ is of positive weight. By Lemma 2.4.4, it follows that $[s,u] \subseteq T_2$, and therefore $[s,u] = \{su\}$ and $T_2 = (s,u,w,t)$. If $z \in V$, then by replacing $wt$ by $zt$ in $S$, we get a solution of $S(G)$. But, as $a(zt) = 0$, this implies that $a(wt) = 0$, a contradiction. Thus $T_1$ cannot go through $V$. In consequence, as by Lemma 2.4.3 $V \neq \emptyset$, there is a node $v \in V$ such that $sv$ and $vt$ do not belong neither to $T_1$ nor to $T_2$. So, by replacing $T_1$ by $(sv, vt)$, we get a solution of $S(G)$. However, since, from Lemma 2.4.10, we have $a(ut) = a(sv)$, we get $a(T_1^{su}) = 0$, a contradiction. Consequently, $[s, u] \cap T_1 \neq \emptyset$ and $T_1 = (s, u, t)$. By using similar arguments, we can also show that $T_2$ is of the form $(f, uw, wt)$, where $f$ is an edge parallel to $su$, and hence $|[s, u]| \geq 2$. Furthermore, at least one of the sets $[u, w]$ and $[w, t]$ is reduced to a single edge. If not, one may replace $ut$ by a 2-$ut$-path going through $w$ and get a solution of $S(G)$. But this would imply that $a(wt) \geq a(ut)$, a contradiction.

Suppose that $|[w, t]| = 1$. The case where $|[u, w]| = 1$ is similar. Hence $[w, t] = \{wt\}$. Let $S' \in S_a(G)$ such that $wt \notin S'$ and suppose that $S'$ is minimal. If $S'$ contains an edge $e \in [u, w]$, then, as $S'$ is minimal, there must exist in $S'$ a 3-$st$-path $T$ containing $e$. Therefore $T$ is of the form $(s, w, u, t)$. Observe that in this case, the edge set obtained by deleting $ut$ and adding $wt$ is in $S(G)$ and then $a(ut) \leq a(wt)$, a contradiction. Consequently, $[u, w] \cap S' = \emptyset$. Hence, as $|T_2 \cap S'| \leq 1$, there is a 3-$st$-path, say $T_1'$, in $S'$ such that $T_1' \cap T_2 = \emptyset$. By replacing $T_1$ by $T_1'$ in $S$, we get a solution of $S(G)$ and hence $a(T_1') \geq a(T_1)$. Note that only one edge of $[s, u]$ can be used by the second 3-$st$-path of $S'$. Thus one can replace $T_1'$ by $T_2$ in $S'$ and obtain a feasible solution, which yields $a(T_2) \geq a(T_1')$, and therefore $a(T_2) \geq a(T_1)$. But this implies that $a(wt) \geq a(ut)$, which is impossible. ♦

Suppose that $a(ut) > a(wt)$. The case where $a(ut) < a(wt)$ can be treated similarly. Let $S_1$ be a minimal solution of $S_a(G)$ that contains $ut$, and let $P_1$ and $P_1'$ be two edge-disjoint 3-$st$-paths of $S_1$. Suppose, w.l.o.g., that $ut \in P_1$. By the Claim, we have $[w, t] \cap S_1 = \emptyset$. If $S_1$ contains an edge of $[u, w]$, then there is a 3-$st$-path of $S_1$ of the form $(s, w, u, t)$. However, by removing $ut$ and adding $wt$, we obtain a solution of $S(G)$, yielding $a(wt) \geq a(ut)$, a contradiction. Thus $[u, w] \cap S_1 = \emptyset$. Moreover, if there is an edge $e$ of $[s, u]$ such that $e \notin P_1'$, one can replace $ut$ by $(e, uw, wt)$ and get a solution of $S(G)$. But this implies that $a(wt) \geq a(ut)$, a contradiction. Consequently, we have that $[s, u] \subseteq P_1'$. Hence $[s, u] = \{su\}$ and $P_1 = (s, z, u, t)$ with $z \in N \backslash \{s, t, u, w\}$. Observe that the subpath $P_1^{su}$ of $P_1$ between $s$ and $u$ is of positive weight. If there are two edges $f \in [s, v]$ and $f' \in [v, t]$ such that $f, f' \notin P_1'$, where $v \in V$, then we can replace $P_1$ by the edges $f$ and $f'$ and still have a feasible solution. As by Lemma 2.4.10, $a(f) = a(ut)$, we obtain that $a(P_1^{su}) = 0$, a contradiction. Thus, for every node $v \in V$, the path $P_1'$ must use all the edges of at least one of the sets $[s, v]$ and $[v, t]$. This implies that $V = \{v\}$. Moreover, as $su \in P_1'$, we have that $[s, v] \cap P_1' = \emptyset$, $[v, t] = \{vt\}$ and $P_1' = (s, u, v, t)$.

Let $S_2$ be a solution of $S_a(G)$ that does not contain $su$. Recall that $[s, u] = \{su\}$. Suppose that $S_2$ is minimal. Thus $S_2$ consists of two edge-disjoint 3-$st$-paths, say $P_2$ and $P_2'$. As $|U| = 1$, by Lemma 2.4.9, we have that $\delta(u) \cap S_2 = \emptyset$. If $S_2$ contains an edge $e$ of $[w, t]$, as $a(e) > 0$, $e$ must belong to one of the 3-$st$-paths of $S_2$, say $P_2$. Since $(\{su\} \cup [u, w]) \cap S_2 = \emptyset$, $P_2$ must be of the form $(s, z', w, t)$ where $z' \notin \{s, t, u\}$. Remark that the subpath of $P_2$ between $s$ and $w$ is of positive weight. Hence, by Lemma 2.4.4, $P_2'$ must intersect every 2-$sw$-path going through $u$. But this contradicts the fact that $(\{su\} \cup [u, w]) \cap S_2 = \emptyset$. It then follows that $[w, t] \cap S_2 = \emptyset$. As $|P_1' \cap S_2| \leq 1$, there is a 3-$st$-path in $S_2$, say $P_2$, that does not intersect $P_1'$. Let $P$ be a 3-$st$-path going through the nodes $s, u, w, t$. From Lemma 2.4.1, it follows that $a(P) \geq a(P_1)$. But then we have that $a(wt) \geq a(ut)$, a contradiction. □

## 2.5 Proof of Theorem 2.3.1

In this section, we prove Theorem 2.3.1, that is $P(G, L) = Q(G, L)$ for $L = 2, 3$. For this, we consider an inequality $ax \geq \alpha$ that defines a facet of $P(G, L)$ different from the trivial and the $st$-cut inequalities. We will show that $ax \geq \alpha$ is necessarily an $L$-$st$-path-cut inequality.

**Case 1.** $L = 2$. Let $U, V, W$ be as defined in the previous section. By Lemma 2.4.6, it follows that $W = \emptyset$ and hence each 2-$st$-path uses exactly one edge with a nonzero coefficient. Thus, any solution of $S_a(G)$ contains exactly two edges with a positive coefficient, which are exactly the edges of the 2-path-cut inequality induced by the partition $\{s\}, U, V, \{t\}$. This implies that $ax \geq \alpha$ is the 2-path-cut inequality induced by this partition.

**Case 2.** $L = 3$. Let $U, V, W_1, W_2, W_3, Z$ be as defined in the previous section. We consider two cases.

**Case 2.1.** $W_1 \cup W_3 \cup Z \neq \emptyset$. Let $F_1 = [\{s\} \cup U, Z] \cup [s, W_1] \cup [U, W_3]$ and $F_2 = [Z, V \cup \{t\}] \cup [W_3, t] \cup [W_1, V]$ (see Figure 2.4). Remark that $F_1 \cap F_2 = \emptyset$ and that there is no $st$-path of length 3 in $G$ formed by edges only from $F_1$ and $F_2$. We have the following.

**Lemma 2.5.1.** *For every solution $S$ of $S_a(G)$, we have that $|S \cap F_1| = |S \cap F_2|$.*

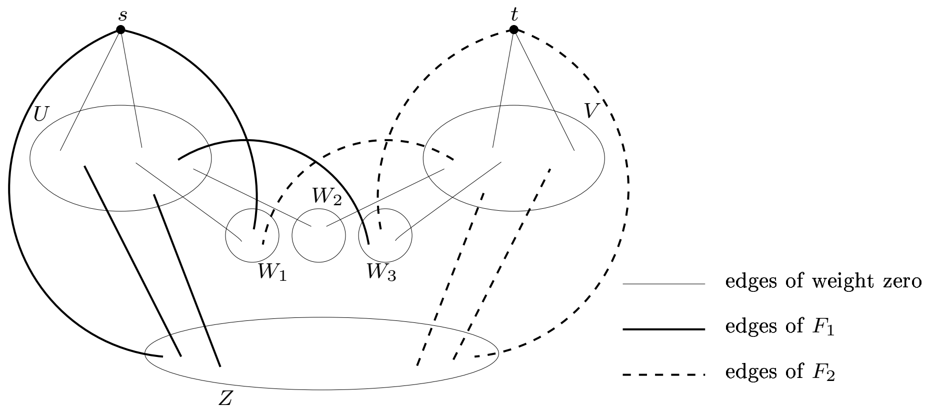


Figure 2.4: The structure generated in Case 2.1

*Proof.* Assume the contrary. Then there exists a solution, say $S_1$, such that, for one of its 3-$st$-path, say $P_1$, we have $|P_1 \cap F_1| \neq |P_1 \cap F_2|$. Let $P_1'$ be the second 3-$st$-path in $S_1$. W.l.o.g., we may suppose that $P_1 \cap F_1 \neq \emptyset$.

*Claim 1.* $P_1 \cap F_2 = \emptyset$.

*Proof.* Since $P_1 \cap F_1 \neq \emptyset$ and $F_1 \cap F_2 = \emptyset$, we have that $|P_1 \cap F_2| \leq 2$. If $|P_1 \cap F_2| = 1$, as $|P_1 \cap F_1| \neq |P_1 \cap F_2|$ and $P_1 \cap F_1 \neq \emptyset$, $|P_1 \cap F_1| = 2$. Then, $P_1$ is of length 3 and contained in $F_1 \cup F_2$, which is impossible by the remark above. If $|P_1 \cap F_2| = 2$, then $|P_1 \cap F_1| = 1$, and again we have that $P_1$ is of length 3 and contained in $F_1 \cup F_2$, a contradiction. Thus, $|P_1 \cap F_2| = 0$ and the claim is proved. ⧫

*Claim 2.* i) $P_1 \cap [s, U] = \emptyset$.
ii) $P_1 = (s, z, w, t)$ with $z \in Z \cup W_1$ and $w \in U \cup W_1 \cup W_2$ ($z$ and $w$ may be the same).
iii) $[s, U] \subset P_1'$.
iv) $|U| = 1$ and $|[s, U]| = 1$.

*Proof.* First note that iv) is a consequence of iii).
i) If $P_1$ uses an edge of $[s, U]$, say $su$ with $u \in U$, as $P_1 \cap F_1 \neq \emptyset$, $P_1$ would be of the form $(s, u, z, t)$ where $z$ belongs to either $Z$ or $W_3$. But this implies that $P_1 \cap F_2 \neq \emptyset$, which contradicts Claim 1.
ii) Suppose that $P_1$ contains an edge of $[U, W_3]$, say $uw_3$. Note that $a(uw_3) > 0$. As, by i), $[s, U] \cap P_1 = \emptyset$, it follows that $P_1 = (s, w_3, u, t)$. By removing $uw_3$ and adding $su$ and edges $w_3v, vt$ for some $v \in V$, we get a solution of $S(G)$. As the added edges

all have zero weight, this implies that $a(uw_3) = 0$, a contradiction. Consequently, we have that $P_1 \cap [U, W_3] = \emptyset$. Then, by i) and the fact that $P_1 \cap F_1 \neq \emptyset$, it follows that $P_1$ uses one of the edges of $[s, Z \cup W_1]$. As, by Claim 1, $P_1 \cap F_2 = \emptyset$, we obtain that $P_1 = (s, z, w, t)$, where $z \in Z \cup W_1$ and $w \in U \cup W_1 \cup W_2$.
iii) Suppose that there is an edge of $[s, U]$, say $su_0$, that does not belong to $P'_1$. We have that $w \neq u_0$. For otherwise, $P_1$ would be $(s, z, u_0, t)$. As by ii) $z \in Z \cup W_1$ and hence $a(sz) > 0$, it follows that the subpath of $P_1$ between $s$ and $u_0$ has a positive weight. But this implies by Lemma 2.4.4 that $su_0 \in P'_1$, a contradiction. We claim that $[u_0, w] \subseteq P'_1$. In fact, if, for instance, $u_0w \notin P'_1$, then consider the solution, say $S'_1$, obtained from $S_1$ by replacing $sz$ and $zw$ by $su_0$ and $u_0w$. Clearly, $S'_1 \in S(G)$, which implies that $a(su_0) + a(u_0w) \geq a(sz) + a(zw)$. As $a(u_0w) = a(su_0) = 0$, we obtain that $a(sz) = 0$, a contradiction. Thus $[u_0, w] \subseteq P'_1$, and hence $[u_0, w] = \{u_0w\}$. Suppose now that $P'_1 = (f, u_0w, g)$ where $f$ (resp. $g$) is an edge of $[s, u_0]$ (resp. $[w, t]$) different from that used by $P_1$. By removing $sz, zw$ and $g$, and adding the edges $su_0$ and $u_0t$, we get a solution of $S(G)$. As by Lemma 2.4.11 $a(u_0t) = a(g)$, it follows that $a(sz) = 0$, a contradiction. Consequently, $P'_1 = (s, w, u_0, t)$. Now, by considering the solution $\tilde{S}_1 = (S_1 \backslash \{sz, zw\}) \cup \{su_0\}$, one can get a contradiction along the same line. This ends the proof of the claim. ♦

Now, by Claim 2 iv), we may suppose that $U = \{u\}$ and $[s, u] = \{su\}$. Let $S_2$ be a solution of $S_a(G)$ that does not contain $su$. W.l.o.g., we may suppose that $S_2$ is minimal. Then, by Lemma 2.4.9, it follows that $S_2 \cap \delta(u) = \emptyset$. Let $P = \{s, u, t\}$. Clearly, $P \cap S_2 = \emptyset$. Moreover, as $P'_1$ goes through node $u$, $|P'_1 \cap S_2| \leq 1$. In consequence, there must exist a 3-$st$-path of $S_2$, say $P_2$, such that $P_2 \cap P'_1 = \emptyset$. Now, by Lemma 2.4.1, we obtain that $a(P) \geq a(P_1)$. By Claim 2 ii) together with Lemma 2.4.11, it follows that $a(sz) \leq 0$. We then have a contradiction and the lemma is proved. □

From Lemma 2.5.1, it follows that the facet defined by $ax \geq \alpha$ is contained in the face induced by the equation $x(F_1) - x(F_2) = 0$. As, by Lemma 2.3.6, this equation cannot be a positive multiple of $ax = \alpha$, we have a contradiction.

**Case 2.2.** $W_1 \cup W_3 \cup Z = \emptyset$. Since by Lemma 2.4.6 $W \neq \emptyset$, we have necessarily that $W_2 \neq \emptyset$. Thus $\{s\}, U, W_2, V, \{t\}$ is a partition of $N$. Let $T$ be the set of edges of the 3-path-cut induced by this partition (these edges are represented by solid lines in Figure 2.5). Note that $a(e) > 0$ for all $e \in T$. Moreover, $a(e) = 0$ for all $e \in E \backslash T$. This is clear for the edges of $E \backslash (T \cup E(W_2))$ from Lemma 2.4.5 ii) and the definition of $U, V, W_2$. If $a(z_1z_2) > 0$ for some $z_1, z_2 \in W_2$, then there must exist a solution $\tilde{S}$ of $S_a(G)$ and a 3-$st$-path $\tilde{P}$ of $\tilde{S}$ containing $z_1z_2$. W.l.o.g., we may suppose that $\tilde{P} = (s, z_1, z_2, t)$. Let

$\tilde{S}' = (\tilde{S} \backslash \{z_1 z_2\}) \cup \{su, uz_2, z_1 v, vt\}$ for some nodes $u \in U$ and $v \in V$. As $\tilde{S}' \in S(G)$ and all the added edges have zero weight, it follows that $a(z_1 z_2) = 0$, a contradiction.

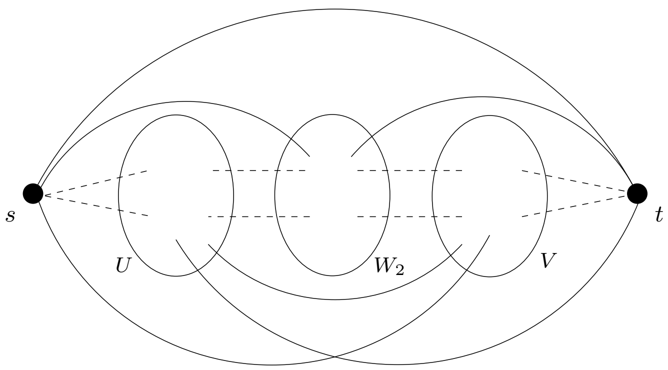


Figure 2.5: The 3-$st$-path-cut generated in Case 2.2

Now we claim that each solution of $S_a(G)$ contains exactly two edges of $T$. First of all, note that, as the constraint (2.3) associated with $T$ is valid for $P(G, 3)$, every solution of $S_a(G)$ must contain at least two edges of $T$. Assume that there is a solution $S$ of $S_a(G)$ with more than two edges of $T$. So, there must exist in $S$ a 3-$st$-path $P$ that contains at least two edges of $T$. We consider the case where $P = (s, w_2, w_2', t)$ with $w_2, w_2' \in W_2$. The other possible cases for $P$ can be treated similarly ($(s, w_2, t)$, $(s, w_2, u, t)$ with $u \in U$, $(s, v, w_2, t)$ with $v \in V$, $(s, v, u, t)$). Let $P'$ be the second 3-$st$-path of $S$. By replacing $P'$ by the edges $su, uw_2', w_2 v, vt$ in $S$, we get a solution of $S(G)$. As all these edges have zero weight, $a(P') = 0$, contradicting Lemma 2.4.2.

Thus, every solution of $S_a(G)$ uses exactly two edges of $T$. This implies that $ax \geq \alpha$ is nothing but the 3-path-cut inequality induced by $T$, which ends the proof of the theorem.

## 2.6 Facets of $P(G, L)$

In this section, we give necessary and sufficient conditions for inequalities (2.1)-(2.3) to be facet-defining for $P(G, L)$. This yields a minimal description of this polytope when $L \leq 3$. Throughout this section, $G = (N, E)$ is a complete graph with $|N| \geq 4$, which may contain multiple edges. Hence, by Corollary 2.3.3, $P(G, L)$ is full dimensional. The first theorem describes when the trivial inequalities define facets of $P(G, L)$.

**Theorem 2.6.1.** *i) For $L \geq 2$, inequality $x(e) \leq 1$ defines a facet of $P(G, L)$.*
*ii) For $L \geq 2$, inequality $x(e) \geq 0$ defines a facet of $P(G, L)$ if and only if $|N| \geq 5$, or $|N| = 4$ and $e$ does not belong to either an $st$-cut or an $L$-$st$-path-cut, with exactly three edges.*

*Proof.* i) Since we are working here on a complete graph of at least four nodes, the edge sets $E$ and $E \backslash \{f\}$, for all $f \in E \backslash \{e\}$, are clearly feasible for the THPP. Moreover, their incidence vectors satisfy the inequality $x(e) \leq 1$ as an equality, and they are affinely independent. Thus, as we have $|E \setminus \{e\}| + 1 = |E|$ such solutions, the face induced by $x(e) \leq 1$ is of dimension $|E| - 1$ and so, is a facet of $P(G, L)$.

ii) Suppose first that $|N| \geq 5$. Then, the four $L$-$st$-paths $\{s, t\}$, $\{s, u, t\}$,$\{s, v, t\}$ and $\{s, w, t\}$ with $u, v, w \in N \backslash \{s, t\}$ are edge-disjoint. Therefore removing any two edges will cut at most two of these four paths. For this reason, the $|E|$ edge sets $E \backslash \{e\}$ and

$$F_f = E \backslash \{e, f\} \quad \text{for all } f \in E \backslash \{e\},$$

all belong to $S(G)$. Moreover, their incidence vectors satisfy $x(e) \geq 0$ with equality and are affinely independent. Hence, $x(e) \geq 0$ defines a facet of $P(G, L)$.

Suppose now that $|N| = 4$. Let $e$ be an edge belonging to an $st$-cut $\delta(W)$ (resp. an $L$-$st$-path-cut $T$) of cardinality three. Then inequality $x(e) \geq 0$ is dominated by the inequalities

$$\begin{aligned} &x(\delta(W)) \geq 2, & & \text{(resp. } x(T) \geq 2\text{)}, \\ &x(f) \leq 1, & \text{for all } f \in \delta(W) \backslash \{e\} \quad & \text{(resp. } f \in T \backslash \{e\}\text{)}, \end{aligned}$$

and hence $x(e) \geq 0$ cannot define a facet.
Now assume that $e$ does not belong to any $st$-cut, or $L$-$st$-path-cut, of cardinality three. By considering the edge sets $E$ and $E \backslash \{e, f\}$ for all $f \in E \backslash \{e\}$, we have $|E|$ solutions of $S(G)$, whose incidence vectors satisfy $x(e) \geq 0$ with equality and are affinely independent. This implies that $x(e) \geq 0$ is facet-defining. □

We now investigate the conditions under which inequalities (2.1) are facet-defining.

**Theorem 2.6.2.** *i) If $L = 2$, then the only $st$-cut inequalities that define facets of $P(G, 2)$ are those induced by $\{s\}$ and $N \setminus \{t\}$.*
*ii) For $L \geq 3$, every $st$-cut inequality defines a facet of $P(G, L)$.*

*Proof.* i) Let $\delta(W)$ be an $st$-cut. Suppose first that $|W| \geq 2$ and $|\overline{W}| \geq 2$. Consider the partition

$V_0 = \{s\}$,
$V_1 = W \backslash \{s\}$,
$V_2 = \overline{W} \backslash \{t\}$,
$V_3 = \{t\}$,

and denote by $T$ the 2-$st$-path-cut induced by this partition. It is easy to see that $T \subsetneq \delta(W)$. Hence, the $st$-cut inequality $x(\delta(W)) \geq 2$ is dominated by the 2-$st$-path-cut inequality $x(T) \geq 2$, along with the nonnegativity constraints.
Thus the only $st$-cut inequalities that may induce facets of $P(G,2)$ are those induced by $\{s\}$ and $N \backslash \{t\}$. In what follows, we shall show that the $st$-cut inequality associated with $W = \{s\}$ is facet-defining. The proof for $W = N \backslash \{t\}$ is similar.

Denote by $ax \geq \alpha$ the inequality $x(\delta(s)) \geq 2$. Let $bx \geq \beta$ be a facet-defining inequality of $P(G,2)$ such that

$$\{x \in P(G,2) \,|\, ax = \alpha\} \subseteq \{x \in P(G,2) \,|\, bx = \beta\}.$$

To show that $ax \geq \alpha$ defines a facet of $P(G,2)$, it suffices to show that $b = \rho a$ for some $\rho > 0$.

First, consider the edge sets

$F_e = \{st, e\} \cup E(\overline{W}), \quad \text{for all } e \in \Delta_{st},$

where $st$ is some edge of $[s,t]$ and $\Delta_{st} = \delta(s) \backslash \{st\}$. As $G$ is complete, $F_e$ belongs to $S(G)$ for all $e \in \Delta_{st}$. Moreover, we have that $ax^{F_e} = \alpha$ for all $e \in \Delta_{st}$. Thus

$0 = bx^{F_e} - bx^{F_f} = b(e) - b(f), \quad \text{for all } e, f \in \Delta_{st},$

and hence,

$$b(e) = \gamma, \quad \text{for all } e \in \Delta_{st} \text{ for some } \gamma \in I\!R.$$

Now, as $|N| \geq 4$, let $u, v \in \overline{W} \backslash \{t\}$ and consider the solution $F = \{su, sv\} \cup E(\overline{W})$, where $su$ and $sv$ are some edge from $[s,u]$ and $[s,v]$, respectively. Obviously, $F \in S(G)$ and $ax^F = \alpha$. As $su \in \Delta_{st}$, and hence $ax^{F_{su}} = \alpha$, we obtain that

$0 = bx^{F_{su}} - bx^F = b(st) - \gamma.$

This yields

$$b(e) = \gamma, \quad \text{for all } e \in \delta(s).$$

Now, we show that $b(f) = 0$ for all $f \in E(\overline{W})$. Let $f$ be such an edge and $u, v$ be its end nodes in $\overline{W}$. If $u \neq t \neq v$, consider the edge set $F'_{su} = F_{su} \backslash \{f\}$. Clearly, this one belongs to $S(G)$ and its incidence vector satisfies $ax \geq \alpha$ with equality. Hence,

$0 = bx^{F_{su}} - bx^{F'_{su}} = b(f)$.

Now, if $v = t$ (the case where $u = t$ is similar), since $|N| \geq 4$, there is a node $w \in N' \backslash \{u, t\}$. Consider the edge set $F'_{sw} = F_{sw} \backslash \{f\}$. Clearly, this one belongs to $S(G)$ and its incidence vector satisfies $ax \geq \alpha$ with equality. Hence,

$0 = bx^{F_{sw}} - bx^{F'_{sw}} = b(f)$.

Altogether, we finally have

$$\begin{aligned} b(e) &= \gamma, & &\text{for all } e \in \delta(s), \\ b(e) &= 0, & &\text{for all } e \in E(\overline{W}). \end{aligned}$$

As $\alpha > 0$ by Lemma 2.3.6, $\gamma > 0$. By setting $\rho = \gamma$, we get $b = \rho a$ and the proof is complete for $L = 2$.

ii) Let us denote inequality (2.1) by $ax \geq \alpha$ and let $bx \geq \beta$ be a facet-defining inequality such that

$$\{x \in P(G, L) \mid ax = \alpha\} \subseteq \{x \in P(G, L) \mid bx = \beta\}.$$

As we did before, we will show $b = \rho a$ for some $\rho > 0$.

Let $\Gamma = E(W) \cup E(\overline{W})$ and $\Delta_{st} = \delta(W) \backslash \{st\}$ for some fixed $st \in [s, t]$. Consider the edge sets

$F_e = \Gamma \cup \{st, e\}$, for all $e \in \Delta_{st}$.

As $F_e \in S(G)$ and $ax^{F_e} = \alpha$ for all $e \in \Delta_{st}$, it follows that

$0 = bx^{F_e} - bx^{F_f} = b(e) - b(f)$, for all $e, f \in \Delta_{st}$.

Hence,

$$b(e) = \gamma, \text{ for all } e \in \Delta_{st}, \text{ for some } \gamma \in \mathbb{R}.$$

It remains to show that edge $st$ also has a $b$-coefficient equal to $\gamma$. This is the object of the following claim.

*Claim.* $b(st) = \gamma$.

*Proof.* Suppose first that $|W| = 1$. Since $|N| \geq 4$, there exist two nodes $u, v$ in $\overline{W}\backslash\{t\}$. Let $F = \{su, sv\} \cup E(\overline{W})$. Clearly, $F \in S(G)$ and $ax^F = \alpha$. As $su \in \Delta_{st}$ and $ax^{F_{su}} = \alpha$, it follows that

$$0 = bx^{F_{su}} - bx^F = b(st) - \gamma.$$

If $|W| = 2$ (resp. $|W| \geq 3$), then, by considering the solution $\{ut, sv\} \cup \Gamma$ (resp. $\{ut, vt\} \cup \Gamma$), where $u \in W\backslash\{s\}$, $v \in \overline{W}\backslash\{t\}$ (resp. $u, v \in W\backslash\{s\}$), we obtain along the same line that $b(st) = \gamma$ and the claim is proved. ♦

Next we show that $b(e) = 0$ for all $e \in \Gamma$. We will prove this for $e \in E(W)$, the proof when $e \in E(\overline{W})$ can be done along the same line.
Let $e \in E(W)$. If $e$ is not incident to $s$, it is clear that $F_f\backslash\{e\}$ is a solution of the problem for each $f \in \Delta_{st}$, and hence $b(e) = 0$.
Thus, suppose that $e \in [s, u]$ for some $u \in W$. If $|\overline{W}| = 1$, as $|N| \geq 4$, $W\backslash\{s, u\} \neq \emptyset$. Let $v \in W\backslash\{s, u\}$ and $f \in [v, t]$. As $F_f\backslash\{e\} \in S(G)$, we have that $b(e) = 0$.
Now, if $|\overline{W}| \geq 2$, let $v' \in \overline{W}\backslash\{t\}$. By considering $f' \in [s, v']$, we have that $F_{f'}\backslash\{e\} \in S(G)$ and hence $b(e) = 0$.

Altogether, we have that

$$b(e) = \gamma, \text{ for all } e \in \delta(W),$$
$$b(e) = 0, \text{ for all } e \in \Gamma.$$

As $\alpha > 0$ by Lemma 2.3.6, $\gamma > 0$. Moreover, by setting $\rho = \gamma$, we have that $b = \rho a$ and the theorem is established. □

We give now necessary and sufficient conditions for the $L$-$st$-path-cut inequalities to be facet-defining for $P(G, L)$.

**Theorem 2.6.3.** *For $L \geq 2$, inequality (2.3) defines a facet of $P(G, L)$ if and only if $|V_0| = |V_{L+1}| = 1$.*

*Proof. Necessity.* We will show that $x(T) \geq 2$ does not define a facet of $P(G, L)$ if $|V_0| \geq 2$. The case where $|V_{L+1}| \geq 2$ follows by symmetry.
Suppose that $|V_0| \geq 2$, and consider the partition given by

$$\overline{V}_0 = \{s\},$$
$$\overline{V}_1 = V_1 \cup (V_0\backslash\{s\}),$$
$$\overline{V}_i = V_i, \; i = 2, \ldots, L+1.$$

This partition induces the $L$-$st$-path-cut inequality $x(\overline{T}) \geq 2$, where $\overline{T} = T\backslash[V_0\backslash\{s\}, V_2]$.

As $G$ is complete, we have that $\overline{T}$ is strictly contained in $T$, and hence, $x(T) \geq 2$ cannot be facet-defining.

*Sufficiency.* Now, suppose that $|V_0| = |V_{L+1}| = 1$, that is $V_0 = \{s\}$ and $V_{L+1} = \{t\}$. Let us denote inequality (2.3) by $ax \geq \alpha$ and let $bx \geq \beta$ be a facet-defining inequality of $P(G, L)$ such that

$$\{x \in P(G, L) \,|\, ax = \alpha\} \subseteq \{x \in P(G, L) \,|\, bx = \beta\}.$$

We will show that $a = \rho b$ for some $\rho > 0$.

Let $V_0 = \{s\}, V_1, \ldots, V_L, V_{L+1} = \{t\}$ be the partition inducing $ax \geq \alpha$. Let $\overline{E} = E \backslash T = (\bigcup_{i=1}^{L} E(V_i)) \cup (\bigcup_{i=0}^{L} [V_i, V_{i+1}])$. Let $f \in [s, t]$ and $T_f = T \backslash \{f\}$. As the graph $G$ is complete, it is easy to see that the sets given by

$$F_e = \overline{E} \cup \{f, e\}, \text{ for all } e \in T_f,$$

induce solutions of the THPP, whose incidence vectors satisfy $ax \geq \alpha$ with equality. Thus,

$$0 = bx^{F_e} - bx^{F_{e'}} = b(e) - b(e'), \text{ for all } e, e' \in T_f.$$

Hence,

$$b(e) = b(e'), \text{ for all } e, e' \in T_f. \tag{2.5}$$

Now let $g \in [V_0, V_L]$, $g' \in [V_1, V_{L+1}]$ $(g, g' \in T_f)$, and $F^* = \overline{E} \cup \{g, g'\}$. It is obvious that $F^*$ induces a solution whose incidence vector satisfies $ax \geq \alpha$ with equality. Thus $bx^{F^*} - bx^{F_g} = b(g') - b(f) = 0$. This together with (2.5) yields

$$b(e) = \gamma, \text{ for all } e \in T \text{ for some } \gamma \in I\!R.$$

Now, we shall show that $b(e) = 0$ for all $e \in \overline{E}$. Suppose first that $e \in [V_0, V_1]$. Consider an edge $h \in [s, w]$ with $w \in V_2$ and the edge set $F_h \backslash \{e\}$, where $F_h$ is as defined above. It is easy to see that $F_h \backslash \{e\}$ still induces a solution of the THPP whose incidence vector satisfies $ax \geq \alpha$ with equality. Thus,

$$0 = bx^{F_h} - bx^{F_h \setminus \{e\}} = b(e).$$

Similarly, we obtain that $b(e) = 0$ for all $e \in \bigcup_{i=0}^{L}[V_i, V_{i+1}]$. Consider now an edge $e \in E(V_i)$, $i \in \{1, \ldots, L\}$. Let $v \in V_L$ and $h' \in [s, v]$. Clearly, the set $F_{h'} \setminus \{e\}$ induces a solution of the problem. As $ax^{F_{h'}} = ax^{F_{h'} \setminus \{e\}} = \alpha$, we have that $bx^{F_{h'}} = bx^{F_{h'} \setminus \{e\}} = \alpha$ and hence $b(e) = 0$.

Consequently, we have that

$$\begin{aligned} b(e) &= 0, \text{ for all } e \in \overline{E}, \\ b(e) &= \gamma, \text{ for all } e \in T. \end{aligned}$$

Since $\alpha > 0$, we have that $\gamma > 0$, and by setting $\rho = 1/\gamma$, we obtain that $a = \rho b$. □

Let $E'$ be the set of edges that do not belong neither to an $st$-cut nor to an $L$-$st$-path-cut, consisting of exactly three edges. From the previous theorems, we have the following.

**Corollary 2.6.4.** *If $G = (N, E)$ is complete and $|N| \geq 4$, then a minimal complete linear description of $P(G, 2)$ is given by*

$$\begin{aligned} & x(\delta(s)) \geq 2, && \\ & x(\delta(t)) \geq 2, && \\ & x(T) \geq 2, && \textit{for all 2-st-path-cut } T \textit{ induced by } V_0 = \{s\}, V_1, V_2, V_3 = \{t\}, \\ & x(e) \leq 1, && \textit{for all } e \in E, \\ & x(e) \geq 0, && \textit{for all } e \in E'. \end{aligned}$$

**Corollary 2.6.5.** *If $G = (N, E)$ is complete and $|N| \geq 4$, then a minimal complete linear description of $P(G, 3)$ is given by*

$$\begin{aligned} & x(\delta(W)) \geq 2, && \textit{for all st-cut } \delta(W), \\ & x(T) \geq 2, && \textit{for all 3-st-path-cut } T \textit{ induced by } V_0 = \{s\}, V_1, V_2, V_3, V_4 = \{t\}, \\ & x(e) \leq 1, && \textit{for all } e \in E, \\ & x(e) \geq 0, && \textit{for all } e \in E'. \end{aligned}$$

## 2.7 Dominant of $P(G,L)$

In this section, we consider the dominant of the polytope $P(G,L)$. We give a complete description of that polyhedron for any graph $G$ and integer $L \geq 2$ such that $P(G,L) = Q(G,L)$.

Let $Dom(P(G,L))$ be the dominant of $P(G,L)$, that is

$$Dom(P(G,L)) = \{y \in I\!R^E \mid \exists\, x \in P(G,L), x \leq y\}.$$

Let $D(G,L)$ be the polyhedron given by

$$\begin{array}{lll}
y(\delta(W)) \geq 2, & \text{for all } st\text{-cut } \delta(W), & \\
y(\delta(W)\backslash\{e\}) \geq 1, & \text{for all } st\text{-cut } \delta(W),\ e \in \delta(W), & (2.6)\\
y(T) \geq 2, & \text{for all } L\text{-path-cut } T, & \\
y(T\backslash\{e\}) \geq 1, & \text{for all } L\text{-path-cut } T,\ e \in T, & (2.7)\\
y(e) \geq 0, & \text{for all } e \in E. & (2.8)
\end{array}$$

**Theorem 2.7.1.** *For every $L \geq 2$, if $P(G,L) = Q(G,L)$, then $Dom(P(G,L)) = D(G,L)$.*

*Proof.* We first prove that $Dom(P(G,L)) \subseteq D(G,L)$. Let $y \in Dom(P(G,L))$. Then there exists $\bar{x} \in P(G,L)$ such that $\bar{x} \leq y$. Hence, $y$ satisfies (2.1), (2.3) and (2.8). We show that $y$ also satisfies constraints (2.6) and (2.7).

Consider a constraint $y(\delta(W)\backslash\{e\}) \geq 1$ of type (2.6). As $\bar{x}(\delta(W)) \geq 2$ and $\bar{x}(e) \leq 1$, we have that

$$\begin{array}{rcl}
y(\delta(W)\backslash\{e\}) & \geq & \bar{x}(\delta(W)\backslash\{e\}) \\
& = & \bar{x}(\delta(W)) - \bar{x}(e) \\
& \geq & 2 - \bar{x}(e) \\
& \geq & 1.
\end{array}$$

Now, in a similar way, we obtain that $y(T\backslash\{e\}) \geq 1$ for all $L$-$st$-path-cut $T$ and $e \in T$. Therefore $Dom(P(G,L)) \subseteq D(G,L)$.

Next we prove that $D(G,L) \subseteq Dom(P(G,L))$. To this end, first let us note that the dominant of $D(G,L)$, $Dom(D(G,L))$, is $D(G,L)$ itself. Thus, to prove that $D(G,L) \subseteq Dom(P(G,L))$, it is sufficient to show that any extreme point $\bar{y}$ of $D(G,L)$ belongs to $P(G,L)$. Indeed, suppose that this is the case. Then any convex combination of extreme points of $D(G,L)$ is also in $P(G,L)$. On the other hand, since $Dom(D(G,L)) = D(G,L)$, any solution $y \in D(G,L)$ can be seen as $\tilde{y} + z$, where $\tilde{y}$ belongs to the convex hull of the extreme points of $D(G,L)$ and $z \geq 0$. As $\tilde{y} \in P(G,L)$, we have therefore that $y \in Dom(P(G,L))$.

So let $\bar{y}$ be an extreme point of $D(G,L)$. As $P(G,L) = Q(G,L)$ and all inequalities in $Q(G,L)$ are in $D(G,L)$ except $x(e) \leq 1, e \in E$, in order to show that $\bar{y} \in P(G,L)$, it suffices to show that $\bar{y}(e) \leq 1$ for all $e \in E$.

Suppose that $\bar{y}(e_0) > 1$ for some $e_0 \in E$. Since $\bar{y}$ is an extreme point of $D(G,L)$, there exists at least one constraint among (2.1),(2.6),(2.3),(2.7), involving the variable $y(e_0)$ and that is tight for $\bar{y}$.

If $\bar{y}(\delta(W)\backslash\{f\}) = 1$ with $e_0 \in \delta(W)\backslash\{f\}$, then, clearly, $\bar{y}(e_0) \leq \bar{y}(\delta(W)\backslash\{f\}) = 1$, a contradiction.

If $\bar{y}(\delta(W)) = 2$ with $e_0 \in \delta(W)$, then $\bar{y}(e_0) + \bar{y}(\delta(W)\backslash\{e_0\}) = 2$, and hence $\bar{y}(e_0) = 2 - \bar{y}(\delta(W)\backslash\{e_0\})$. As $\bar{y}$ satisfies (2.6), it follows that $\bar{y}(e_0) \leq 1$, which is impossible.

We obtain a similar contradiction if one of the constraints (2.3),(2.7) is tight for $\bar{y}$. □

It would be interesting to investigate the dominant of the THPP polytope when $P(G,L) \neq Q(G,L)$. An immediate consequence of Theorems 2.3.1 and 2.7.1 is the following.

**Corollary 2.7.2.** *If $L = 2,3$, then $Dom(P(G,L)) = D(G,L)$.*

## 2.8 Concluding remarks

In this chapter, we have considered the problem of finding a minimum cost edge set containing at least two edge-disjoint paths between two terminals $s$ and $t$ of length no more than $L$, where $L \geq 2$ is a given integer. We have given some valid inequalities for this problem and, when $L = 2,3$, an integer programming formulation in the space of the design variables. We have also investigated its polyhedral structure when $L = 2,3$. In particular, we have shown in that case that the associated polytope $P(G,L)$ is described by the trivial, $st$-cut and $L$-$st$-path-cut inequalities. Moreover, we have given necessary and sufficient conditions for these inequalities to be facet-defining for any $L \geq 2$. This yielded a complete and minimal linear description for $P(G, L)$ when $L = 2, 3$. We have finally considered the dominant of $P(G, L)$, for which we have given a complete description for any $L \geq 2$ when $P(G, L)$ is given by those inequalities.

Note that the results for $L = 2$ could be easily extended to the node version of the THPP, that is, when the two required $L$-$st$-paths must be node-disjoint. For this new problem, one can indeed suppose that the underlying graph $G$ is simple. Therefore, any two edge-disjoint 2-$st$-paths are also node-disjoint, and the two problems are equivalent.

Since the separation problems for inequalities (2.1) and (2.3) can be solved in polynomial time when $L \leq 3$ (see Section 2.2), from Theorem 2.3.1, it follows that, for $L \leq 3$, the THPP can be solved in polynomial time using a cutting plane algorithm. To the best of our knowledge, this is the first (non-enumerative) polynomial algorithm devised for this problem. Since a more general cutting plane algorithm can be derived for the problem where an arbitrary number $k \geq 2$ of edge-disjoint $L$-$st$-paths is required ($k$HPP), we will present computational results for it later on, in Chapter 3.

Moreover, a natural question that may be posed is whether a similar integer programming formulation can be obtained for the $k$HPP (when $L = 2, 3$), and whether its linear relaxation is still integral. This motivates us to give the following conjecture.

**Conjecture 2.8.1.** *When $L = 2, 3$, the linear relaxation of $k$HPP is integral for any $k \geq 2$.*

In Chapter 3, we will show that this conjecture holds true when $L = 2$. The question for $L = 3$ will remain open, although the computational experiments will give us some empirical idea about the answer.

For $L = 2, 3$, the formulation given in Section 2.2 for the THPP can be extended to the case where more than one pair of terminals is considered. However, the linear system of constraints will no longer be sufficient to completely describe the associated polytope. In fact, consider the graph shown in Figure 2.6 with two pairs of terminals $\{s, t\}$ and $\{s', t'\}$. Suppose that $L = 2$. Here, a feasible solution must contain at least two edge-disjoint 2-$st$-paths and at least two edge-disjoint 2-$s't'$-paths. It is not hard to see that the fractional point $\bar{x} = (1, 1, 1, 1, 0, 0, 0, 1/2, 1/2, 1/2)$ satisfies all trivial, $st$-cut and $L$-$st$-path-cut inequalities (with respect to the two pairs of terminals).

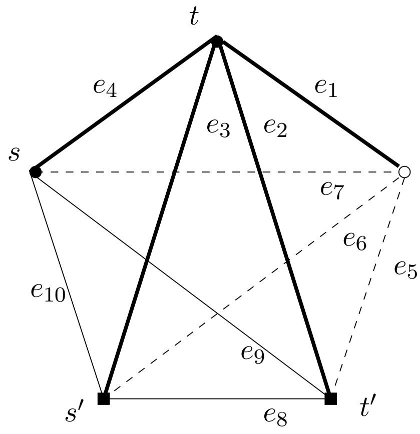


Figure 2.6: A fractional extreme point for the problem with several demands

Moreover, $\bar{x}$ is an extreme point of the polyhedron given by these inequalities. Actually, the following inequality

$$x(e_3) + x(e_5) + x(e_7) + x(e_8) + x(e_9) + x(e_{10}) \geq 3 \tag{2.9}$$

is valid for the problem, but violated by $\bar{x}$. We will study in depth this more general problem in Chapters 4 and 5. In fact, inequality (2.9) will be nothing but a double path-cut inequality (see Section 5.2).

Finally, as it has already been mentioned, if $L \geq 4$, the formulation given in Section 2.2 is no longer valid for the THPP. This question will be considered in Chapter 6.

# Chapter 3

# $k$ Edge-Disjoint Hop-Constrained Paths Problem

In this chapter, we consider the $k$-edge connected $L$-hop-constrained network design problem for $D = \{s, t\}$. This problem, called the $k$ *edge-disjoint Hop-constrained Paths Problem* (or $k$HPP for short), consists in finding a minimum cost subgraph containing at least $k$ edge-disjoint $st$-paths of at most $L$ hops with $k, L \geq 2$. It can also be seen as an extension of the THPP (see Chapter 2) to any value $k$. In particular, we will prove that Conjecture 2.8.1 holds true when $L = 2$. This result has been the object of an article in collaboration with Geir Dahl, A. Ridha Mahjoub and Pierre Pesneau [18]. Moreover, we have added some computational experiments of a cutting plane algorithm (and of an enumerative procedure) to solve the $k$HPP when $L = 2, 3$. These will also give some practical insight about the validity of Conjecture 2.8.1 for $L = 3$.

## 3.1 Introduction

Given a function $c : E \rightarrow I\!R$ which associates a cost $c(e)$ to each edge $e \in E$, the $k$ *edge-disjoint Hop-constrained Paths Problem* ($k$HPP) is to find a minimum cost subgraph such that between $s$ and $t$ there exist at least $k$ edge-disjoint $L$-$st$-paths.

In this chapter, we consider the $k$HPP for $L = 2, 3$. We give an integer programming formulation for this problem and discuss its associated polytope. In particular, we give a minimal complete linear description of that polytope for $L = 2$. Finally, we present some computational results of a cutting plane algorithm to solve the $k$HPP when $L = 2, 3$. We compare these results with those obtained thanks to an enumerative algorithm. Also, these tests will give us some partial answer about the previous linear description being or not complete for $L = 3$.

The convex hull of the incidence vectors of the solutions to the $k$HPP on $G$, denoted by $P_k(G, L)$, will be called the $k$HPP *polytope*. If $x^F$ is the incidence vector of the edge set $F$ of a solution to the $k$HPP, then clearly $x^F$ satisfies the following inequalities

$$x(\delta(W)) \geq k, \quad \text{for all } st\text{-cut } \delta(W), \tag{3.1}$$

$$x(T) \geq k, \quad \text{for all } L\text{-}st\text{-path-cut } T, \tag{3.2}$$

$$1 \geq x(e) \geq 0, \quad \text{for all } e \in E. \tag{3.3}$$

Inequalities (3.1) will be called *st-cut inequalities*, inequalities (3.2) *L-(st-)path-cut inequalities*, and inequalities (3.3) *trivial inequalities*. Remark that these are the generalization to any $k \geq 2$ of the inequalities (2.1)-(2.3) introduced in the previous chapter. Their validity can be shown along the same line as in Theorem 2.2.1.

Now, let $L = 2$ and consider Figure 3.1 for an example of a 2-path-cut inequality with $V_0 = \{s\}$ and $V_3 = \{t\}$.

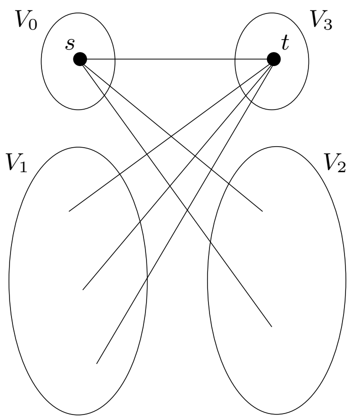


Figure 3.1: Support graph of a 2-path-cut inequality

Remark that the 2-path-cut $T$ intersects each 2-$st$-path in exactly one edge. Let $E_1$ be the set of edges involved in a 2-$st$-path in $G$. Thus, $E_1$ consists of the edges $[s, t]$ and $[s, v]$, $[v, t]$ for all those nodes $v$ for which $G$ contains these edges. Let $G_1 = (N, E_1)$ be the subgraph of $G$ induced by $E_1$.

Observe that, when $L = 2$, it is equivalent to consider the $k$HPP on $G$ and on $G_1$. More precisely, an optimal solution in $G$ will consist of an optimal solution in $G_1$, plus the edges in $E \backslash E_1$ of negatives costs, if any.

Also, it is not hard to see that $T$ (in $G$) corresponds to the $st$-cut $\delta(V_0 \cup V_1)$ in $G_1$. Therefore, we can consider the inequalities

$$x(\delta_{G_1}(W)) \geq k, \text{ for all } W \subset N, s \in W, t \notin W, \tag{3.4}$$

where $\delta_{G_1}(W)$ stands for the cut induced by $W$ in $G_1$. Clearly, inequalities (3.4) dominate inequalities (3.1) and (3.2).

Let $Q_k(G, L)$ be the solution set of the system given by inequalities (3.1)-(3.3). In the next section, we show that inequalities (3.1)-(3.3), together with the integrality constraints, give an integer programming formulation for the $k$HPP when $L = 2, 3$. Moreover, we show that one can also formulate the $k$HPP when $L = 2$ using only inequalities (3.3) and (3.4), along with the integrality constraints. In Section 3.3, we study the $k$HPP polytope for $L = 2$, $P_k(G, 2)$, and show that $P_k(G, 2)$ is completely described by these latter two classes of linear inequalities. In Section 3.4, we discuss the polynomial time solvability of the problem and give some computational results of a cutting plane algorithm for $L = 2, 3$. We also test an enumerative procedure to solve the problem. Finally, in Section 3.5, we propose some concluding remarks.

## 3.2 Formulation for $L = 2, 3$

In this section, we show that the $st$-cut, $L$-$st$-path-cut and trivial inequalities, together with the integrality constraints, suffice to formulate the $k$HPP with $L = 2, 3$ as a $0-1$ linear program. To this aim, we first give a lemma. The proof is omitted because it is along the same line as the proof of Lemma 2.2.2.

**Lemma 3.2.1.** *Let $G = (N, E)$ be a graph, $s$, $t$ be two nodes of $N$, and $L \in \{2, 3\}$. Suppose that there do not exist $k$ edge-disjoint $L$-$st$-paths in $G$, with $k \geq 2$. Then there exists a set of at most $k - 1$ edges that intersects every $L$-$st$-path (if any).*

**Theorem 3.2.2.** *Let $G = (N, E)$ be a graph, $k \geq 2$, and $L \in \{2, 3\}$. Then the $k$HPP is equivalent to the integer program*

$$Min \ \{cx \ : \ x \in Q_k(G, L), \ x \in \mathbb{Z}^E\}.$$

*Proof.* To prove the theorem, it is sufficient to show that every $0-1$ solution $x$ of $Q_k(G,L)$ induces a solution of the $k$HPP when $L=2,3$. Let us assume the contrary. Suppose that $x$ does not induce a solution of the $k$HPP, but that it satisfies the $st$-cut and trivial constraints. We will show that $x$ necessarily violates at least one of the $L$-$st$-path-cut constraints $x(T)\geq k$.

Let $G_x$ be the subgraph induced by $x$. As $x$ is not a solution of the problem, $G_x$ does not contain $k$ edge-disjoint $L$-$st$-paths. It then follows, by Lemma 3.2.1, that there exists a set of at most $k-1$ edges in $G_x$ that intersects every $L$-$st$-path. Consider the graph $\tilde{G}_x$ obtained from $G_x$ by deleting these edges. Obviously, $\tilde{G}_x$ does not contain any $L$-$st$-path. We claim that $\tilde{G}_x$ contains at least one $st$-path of length at least $L+1$. In fact, as $x$ is a $0-1$ solution and satisfies the $st$-cut inequalities, $G_x$ contains at least $k$ edge-disjoint $st$-paths. Since at most $k-1$ edges were removed from $G_x$, at least one path remains between $s$ and $t$ in $\tilde{G}_x$. However, since $\tilde{G}_x$ does not contain an $L$-$st$-path, that path must be of length at least $L+1$.

Now consider the partition $V_0,\dots,V_{L+1}$ of $N$, with $V_0=\{s\}$, $V_i$ the set of nodes at distance $i$ from $s$ in $\tilde{G}_x$, for $i=1,\dots,L$, and $V_{L+1}=N\setminus\left(\bigcup_{i=0}^{L}V_i\right)$, where the distance between two nodes is the length of a shortest path between these nodes. Since there does not exist an $L$-$st$-path in $\tilde{G}_x$, it is clear that $t\in V_{L+1}$. Moreover, as, by the claim above, $\tilde{G}_x$ contains an $st$-path of length at least $L+1$, the sets $V_1,\dots,V_L$ are nonempty. Furthermore, no edge of $\tilde{G}_x$ is a chord of the partition (that is, an edge between two sets $V_i$ and $V_j$, where $|i-j|>1$). In fact, suppose that there exists an edge $e=v_iv_j\in[V_i,V_j]$ with $|i-j|>1$ and $i<j$. Then $v_j$ is at distance $i+1<j$ from $s$, a contradiction.

Thus, the edges deleted from $G_x$ are the only edges that may be chords of the partition in $G_x$. In consequence, if $T$ is the set of chords of the partition in $G$, then $x(T)\leq k-1$. But this implies that the corresponding $L$-$st$-path-cut inequality is violated by $x$. □

Of course, since for $k=2$ we already had that this formulation was no more sufficient when $L\geq 4$, the same problem also occurs for the $k$HPP. On the other hand, it is easy to show that, when $L=2$, the problem can be advantageously formulated using inequalities (3.4) instead of (3.1) and (3.2).

**Theorem 3.2.3.** *Let $G=(N,E)$ be a graph and $k\geq 2$. Then the $k$HPP with $L=2$ is equivalent to the integer program*

$$\textit{Min}\ \{cx : x \textit{ satisfies (3.3) and (3.4)},\ x\in\mathbb{Z}^E\}.$$

*Proof.* To prove the theorem, it is sufficient to show that every $0-1$ solution $x$ satisfying (3.3),(3.4) induces a solution of the $k$HPP for $L=2$. Let us assume the contrary.

Suppose that $x$ does not induce a solution of the $k$HPP for $L = 2$. If $x$ does not satisfy an $st$-cut inequality, then clearly one of the inequalities (3.4) is not satisfied. So suppose that $x$ satisfies the $st$-cut and trivial constraints. Then, by doing the same proof as in the previous theorem, one can show that $x$ necessarily violates at least one inequality (3.4) that corresponds to a 2-path-cut constraint. □

## 3.3 Facets and Completeness

In this section, we will show that inequalities (3.3),(3.4) completely describe the polytope $P_k(G,2)$. This answers Conjecture 2.8.1 when $L = 2$. In order to give a minimal system for this polytope, we first study when these inequalities are facet-defining.

### 3.3.1 Facets

We first establish the dimension of $P_k(G,L)$. As for $k = 2$ (see Section 2.3), we say that an edge $e \in E$ is $L$-$st$-essential if $e$ belongs either to an $st$-cut of cardinality $k$, or to an $L$-$st$-path-cut of cardinality $k$. Let $E^*$ denote the set of $L$-$st$-essential edges. Thus, $P_k(G - e, L) = \emptyset$ for all $e \in E^*$. We have the following theorem, which can be easily seen true along the same line as in Theorem 2.3.2.

**Theorem 3.3.1.** $\dim(P_k(G,L)) = |E| - |E^*|$.

Throughout this section, $G = (N,E)$ is a complete graph with $|N| \geq k+2$, which may contain multiple edges. Hence any $st$-cut and $L$-$st$-path-cut of $G$ contains at least $k+1$ edges. Therefore, by Theorem 3.3.1, $P_k(G,L)$ is full dimensional.

**Theorem 3.3.2.** *i) For $L \geq 2$, inequality $x(e) \leq 1$ defines a facet of $P_k(G,L)$.*
*ii) For $L \geq 2$, inequality $x(e) \geq 0$ defines a facet of $P_k(G,L)$ if and only if $|N| \geq k+3$, or $|N| = k+2$ and $e$ does not belong to either an $st$-cut or an $L$-$st$-path-cut with exactly $k+1$ edges.*

*Proof.* i) As $|N| \geq k+2$, and $P_k(G,L)$ is full dimensional, $E_f = E \backslash \{f\}$ is a solution for the $k$HPP for every $f \in E\backslash\{e\}$. Hence the sets $E$ and $E_f$ for $f \in E\backslash\{e\}$ constitute a family of $|E|$ solutions of the $k$HPP. Moreover, their incidence vectors satisfy $x(e) = 1$, and are affinely independent.

ii) Suppose first that $|N| \geq k+3$. Then $G$ contains $k+2$ edge-disjoint $L$-$st$-paths (an edge of $[s,t]$ and $k+1$ paths of the form $(s,u,t)$, $u \in N\backslash\{s,t\}$). Hence any edge set $E\backslash\{f,g\}$, $f,g \in E$, contains at least $k$ paths among these $k+2$ $L$-$st$-paths. Consider the sets

$$E_f = E\backslash\{e,f\} \text{ for all } f \in E\backslash\{e\}.$$

By the above remark, these sets, along with $E\backslash\{e\}$, induce solutions of the $k$HPP. Now, it is easy to see that the incidence vectors of $E\backslash\{e\}$ and $E_f$, $f \in E\backslash\{e\}$, all satisfy $x(e) = 0$ and are affinely independent.
Now suppose that $|N| = k+2$. If $e$ belongs to an $st$-cut $\delta(W)$ (resp. an $L$-$st$-path-cut $T$) with $k+1$ edges, then $x(e) \geq 0$ is redundant with respect to the inequalities

$$\begin{array}{lll} x(\delta(W)) \geq k, & & (\text{resp. } x(T) \geq k), \\ -x(f) \geq -1, & \text{for all } f \in \delta(W)\backslash\{e\}, & (\text{resp. } f \in T\backslash\{e\}), \end{array}$$

and hence, cannot be facet-defining.
If $e$ does not belong to neither an $st$-cut nor an $L$-$st$-path-cut with $k+1$ edges, then the edge sets $E\backslash\{e\}$ and $E_f$, $f \in E\backslash\{e\}$, introduced above, are still solutions for the problem. Moreover, their incidence vectors satisfy $x(e) = 0$ and are affinely independent. □

Now we show that inequalities (3.4) are always facet-defining.

**Theorem 3.3.3.** *Constraints (3.4) define facets for $P_k(G,2)$.*

*Proof.* Let us denote inequality (3.4) by $ax \geq \alpha$, and let $bx \geq \beta$ be a facet defining inequality of $P_k(G,2)$ such that

$$\{x \in P_k(G,2) \,|\, ax = \alpha\} \subseteq \{x \in P_k(G,2) \,|\, bx = \beta\}.$$

We will show that $b = \rho a$ for some $\rho > 0$. Let $T = \delta_{G_1}(W)$, $W_1 = W\backslash\{s\}$ and $W_2 = \overline{W}\backslash\{t\}$. Let $\overline{E} = E\backslash T$. As $|N| \geq k+2$, $|W_1| + |W_2| \geq k$. So consider $k$ nodes $v_1, \ldots, v_k \in W_1 \cup W_2$. Suppose that $v_1, \ldots, v_q \in W_1$ and $v_{q+1}, \ldots, v_k \in W_2$ for some $0 \leq q \leq k$. Let $e_i \in [v_i, t]$ for $i = 1, \ldots, q$ and $e_i \in [s, v_i]$ for $i = q+1, \ldots, k$. Let
$F_1 = \{e_1, \ldots, e_k\} \cup \overline{E}$.
It is clear that $F_1$ induces a solution of the $k$HPP. Let $e \in T\backslash\{e_1, \ldots, e_k\}$. If $e$ is

parallel to one of the edges $e_i$, say $e_1$, then clearly $F_1' = (F_1 \backslash \{e_1\}) \cup \{e\}$ still induces a solution for the problem. Since $ax^{F_1} = ax^{F_1'} = \alpha$, we get $b(e_1) = b(e)$. This implies that

$$b(f) = \rho_i, \text{ for every } f \text{ parallel to } e_i, \text{ for some } \rho_i \in I\!R, \text{ for } i = 1, \ldots, k. \qquad (3.5)$$

If $e$ is not parallel to any $e_i$, then $F_j' = (F_1 \backslash \{e_j\}) \cup \{e\}$ induces a solution for the $k$HPP, for $j = 1, \ldots, k$. As an edge of $[s,t]$ is such an edge, this together with (3.5) implies that, for some $\rho \in I\!R$,

$$b(e) = \rho, \qquad \text{for every edge } f \in T. \qquad (3.6)$$

Now we shall show that $b(f) = 0$ for all $f \in \overline{E}$. Suppose $f \in [s, W_1]$. If $f$ is not incident to any node among $v_1, \ldots, v_q$, then $F_1 \backslash \{f\}$ induces a solution of the problem, and hence, $b(e) = 0$. If $f \in [s, v_i]$ for some $1 \leq i \leq q$, then let

$$\tilde{F}_1 = (F_1 \backslash \{f, e_i\}) \cup \{g\},$$

where $g$ is an edge of $[s,t]$. It is easy to see that $\tilde{F}_1$ still induces a solution of the $k$HPP. As, by (3.6), $b(e_i) = b(g)$, it follows that $b(f) = 0$. Similarly, we can show that $b(f) = 0$ for all $f \in [W_2, t]$. If $f \in [W_1, W_2] \cup E(W_1) \cup E(W_2)$, then obviously $F_1 \backslash \{f\}$ is a solution of the problem, and hence, we obtain that $b(f) = 0$.

Thus we have that

$$\begin{aligned} b(e) &= \rho, && \text{if } e \in T, \\ b(e) &= 0, && \text{if not.} \end{aligned}$$

Since $ax \geq \alpha$ is not a trivial inequality, we have that $\rho > 0$, and hence, that $b = \rho a$. □

### 3.3.2 The polytope $P_k(G, 2)$

We now consider the polytope $P_k(G, 2)$ and we give a complete linear description of this polytope. In fact, we will use an idea from [14] which was used to find a complete linear description of the dominant of $P_1(G, 2)$. The idea is based on the fact that the only edges $e \in E$ that can lie in a 2-$st$-path are $[s,t]$, and $[s,v]$, $[v,t]$ for $v \neq s, t$, that is, the edges of $E_1$. Thus, essentially, the remaining edges play no role. Our proof uses this reduction combined with a well-known result on edge-disjoint paths.

A linear system $Ax \leq b$ is *totally dual integral (TDI)* if, for all integral $c$ such that $\max\{c^T x : Ax \leq b\}$ is finite, the dual $\min\{y^T b : y^T A = c^T, y \geq O\}$ has an integral optimal solution. This implies that, if $Ax \leq b$ is TDI and $b$ is integral, then the polyhedron given by $Ax \leq b$ is integral. In what follows, we give a TDI system that characterizes $P_k(G, 2)$.

**Theorem 3.3.4.** *The system given by (3.3) and (3.4) completely describes the polytope $P_k(G,2)$. Moreover, this system is TDI.*

*Proof.* Observe first that $P_k(G,2)$ is the product of the polytope $P_k(G_1,2)$ and $[0,1]^{E\setminus E_1}$, where $G_1 = (N, E_1)$ is the subgraph of $G$ containing only the edges belonging to 2-$st$-paths (as defined in Section 3.1). Note that in $G_1$ every $st$-path is a 2-$st$-path. Thus, $P_k(G_1,2)$ is the convex hull of incidence vectors of edge sets in $G_1$ containing $k$ edge-disjoint $st$-paths and therefore it equals the solution set of the system

$$\begin{aligned} &x(\delta_{G_1}(W)) \geq k, && \text{for all } W \subseteq N, s \in W, t \notin W, \\ &0 \leq x_e \leq 1, && \text{for all } e \in E. \end{aligned}$$

This is a direct consequence of a well-known result on edge-disjoint paths (a recent reference is Schrijver [61]). Moreover, the system is TDI. The theorem now follows by noting that the TDI property extends to the product of the two polytopes (this is immediate from the definition of TDI). □

Note that, if $G = (N, E)$ is a complete graph, with $|N| \geq k+2$, then, from Theorems 3.3.2, 3.3.3 and 3.3.4, we have the following.

**Corollary 3.3.5.** *If $G = (N, E)$ is complete and $|N| \geq k + 2$, a minimal complete linear system describing $P_k(G,2)$ is the following.*

$$\begin{aligned} &x(\delta_{G_1}(W)) \geq k, && \textit{for all } W \subset N, s \in W, t \notin W, \\ &x(e) \leq 1, && \textit{for all } e \in E, \\ &x_e \geq 0, && \textit{for all } e \in E \textit{ satisfying condition ii) of Theorem 3.3.2.} \end{aligned}$$

In terms of $st$-cut and $L$-path-cut inequalities in $G$, we then have the result here under, which answers Conjecture 2.8.1 for $L = 2$.

**Corollary 3.3.6.** *If $G = (N, E)$ is complete and $|N| \geq k + 2$, a minimal complete linear system describing $P_k(G,2)$ is the following.*

$$\begin{aligned} &x(\delta(s)) \geq k, && \\ &x(\delta(t)) \geq k, && \\ &x(T) \geq k, && \textit{for all 2-st-path-cut } T \textit{ with } V_0 = \{s\} \textit{ and } V_3 = \{t\}, \\ &x(e) \leq 1, && \textit{for all } e \in E, \\ &x_e \geq 0, && \textit{for all } e \in E \textit{ satisfying condition ii) of Theorem 3.3.2.} \end{aligned}$$

## 3.4 Solvability and computational results

The separation problem for inequalities (3.4) can be solved in polynomial time using any polynomial max-flow algorithm (e.g. [44]). Therefore the $k$HPP for $L = 2$ can be solved in polynomial time using a cutting plane algorithm.

Also note that this polynomial cutting plane can be used for solving the node-disjoint case, that is to find a minimum cost subgraph containing at least $k$ node-disjoint 2-$st$-paths. In fact, for this problem, we can suppose that the underlying graph does not contain multiple edges. In consequence, as $L = 2$, two $L$-$st$-paths are node-disjoint if and only if they are edge-disjoint. Therefore, the system given before is also a complete minimal description of the associated polytope.

The separation problem for the $L$-$st$-path-cut inequalities (3.2) can be solved in polynomial time if $L \leq 3$ [25]. Hence these inequalities can be used, together with inequalities (3.1), within the framework of a polynomial time cutting plane algorithm for the $k$HPP for $L = 2, 3$. However, when $L = 3$, we do not have the guarantee that this procedure will solve the problem to optimality. Indeed, except for $k = 2$ (see Chapter 2), we do not know if this linear system is integral or not.

Nevertheless, we will now present computational results of such a cutting plane algorithm for the $k$HPP, not only for $L = 2$, but also for $L = 3$. In the latter case, whether or not we will always obtain a feasible solution will at least gives us some insight about the linear relaxation being still integral or not.

The computational results presented here concern instances consisting in complete graphs with edge costs equal to random values in the interval $[1, 1000]$. These random $k$HPP problems were generated with $n = 20$ to $n = 100$ nodes, for different numbers $k$ of required edge-disjoint $L$-$st$-paths. For each couple $(n, k)$, five instances were tested for $L = 2$, and five others for $L = 3$.

The cutting plane algorithm has been implemented in C++, using CPLEX 8.11 as linear solver, and tested on a Pentium III at 933 MHz with 384 Mo of RAM under Linux. The separations for the $st$-cut inequalities and $L$-$st$-path-cut inequalities are performed exactly as described in Section 2.2. Recall that for $L = 3$ and $k > 2$ we do not have the guarantee that this algorithm will reach an optimal integer solution. We have indeed not established that those two families of inequalities, along with the trivial constraints, are sufficient to describe the $k$HPP polytope in that case.

In the following table, the entries are:

| | | |
|---|---|---|
| $n$ | : | the number of nodes in the graph, |
| $k$ | : | the number of required edge-disjoint $L$-$st$-paths, |
| Cut2 | : | the average number of $st$-cut inequalities generated over the five random $k$HPP instances tested when $L = 2$, |
| Pcut2 | : | the average number of 2-$st$-path-cut inequalities, |
| CPU2 | : | the average CPU time in seconds when $L = 2$, |
| Cut3 | : | the average number of $st$-cut inequalities generated over the five random $k$HPP instances tested when $L = 3$, |
| Pcut3 | : | the average number of 3-$st$-path-cut inequalities, |
| CPU3 | : | the average CPU time in seconds when $L = 3$. |

In Table 3.1, we see that all instances have been solved within a very reasonable runtime. For $L = 2$, the maximum is about 5 seconds for instances with 100 nodes and 45 paths, while, for $L = 3$, it is about 3.5 minutes for instances with 100 nodes and 78 paths. Moreover, for all $k$HPP instances tested when $L = 3$, the resulting optimal solution is integer. This lets us think that Conjecture 2.8.1 also holds true for $L = 3$. In most cases, we remark that the numbers of generated $st$-cut inequalities and of $L$-$st$-path-cut inequalities are quite similar, slightly in favor of the first ones, maybe because they are separated before the second ones. Finally, it seems that the computing time increases from $k = 2$ to $k = n/2$, and decreases from $k = n/2$ to $k = n - 1$. This general behavior is confirmed through the following two graphs, giving the CPU time in function of $k$ for a fixed instance of 100 nodes, for $L = 2$ and $L = 3$, respectively.

In Figure 3.2, we see that the maximum runtime when $L = 2$ is attained around 45 edge-disjoint paths. Moreover, the slope of the curve mainly keeps the same sign before and after this critical value for $k$. On the contrary, when $L = 3$, the curve in Figure 3.3 presents several local maxima. Despite this, the greatest runtime is still reached around $k = n/2$, namely for 55 edge-disjoint paths.

Note that, if the graph has positive costs and no parallel edges, the problem for

Table 3.1: Results for random $k$HPP instances when $L = 2$ and $L = 3$

| $n$ | $k$ | Cut2 | Pcut2 | CPU2 | Cut3 | Pcut3 | CPU3 |
|---|---|---|---|---|---|---|---|
| 20 | 2 | 11 | 9 | 0.028 | 23 | 3 | 0.049 |
| 20 | 5 | 7 | 7 | 0.027 | 19 | 8 | 0.067 |
| 20 | 10 | 10 | 10 | 0.034 | 19 | 11 | 0.084 |
| 20 | 15 | 9 | 9 | 0.034 | 23 | 4 | 0.095 |
| 20 | 19 | 0 | 0 | 0.011 | 0 | 0 | 0.010 |
| 40 | 2 | 22 | 13 | 0.073 | 54 | 9 | 0.229 |
| 40 | 10 | 22 | 22 | 0.124 | 52 | 49 | 0.825 |
| 40 | 19 | 22 | 22 | 0.139 | 55 | 46 | 0.873 |
| 40 | 30 | 15 | 15 | 0.111 | 57 | 31 | 0.851 |
| 40 | 39 | 0 | 0 | 0.022 | 0 | 0 | 0.023 |
| 60 | 2 | 44 | 26 | 0.241 | 84 | 12 | 0.614 |
| 60 | 15 | 40 | 40 | 0.454 | 79 | 75 | 4.269 |
| 60 | 30 | 41 | 41 | 0.563 | 137 | 133 | 9.929 |
| 60 | 45 | 24 | 24 | 0.326 | 144 | 117 | 9.097 |
| 60 | 59 | 0 | 0 | 0.040 | 0 | 0 | 0.041 |
| 80 | 2 | 21 | 17 | 0.254 | 112 | 5 | 1.537 |
| 80 | 19 | 50 | 50 | 1.208 | 245 | 243 | 35.746 |
| 80 | 39 | 62 | 62 | 1.815 | 270 | 267 | 45.783 |
| 80 | 59 | 33 | 33 | 0.857 | 159 | 152 | 21.950 |
| 80 | 78 | 9 | 9 | 0.268 | 26 | 0 | 0.727 |
| 100 | 2 | 39 | 22 | 0.530 | 217 | 7 | 8.190 |
| 100 | 19 | 53 | 53 | 1.964 | 503 | 500 | 152.323 |
| 100 | 45 | 84 | 84 | 4.730 | 493 | 491 | 181.618 |
| 100 | 78 | 44 | 44 | 1.734 | 553 | 537 | 207.464 |
| 100 | 98 | 7 | 7 | 0.348 | 87 | 0 | 4.764 |

$L = 2$ can also be solved polynomially, for $k$ fixed, by enumerating the (at most) $n - 1$ different 2-$st$-paths in $G$ and picking the $k$ of these paths with smallest cost. The same holds for any fixed values of $L$ and $k$, since the number of $L$-$st$-paths is always polynomial for simple graphs and the edge-disjunction of $k$ $L$-$st$-paths can be verified in polynomial time (by verifying that the resulting incidence vector is in $0 - 1$). Note that our cutting plane algorithm for $L = 2, 3$ is polynomial even when $k$ is not fixed. Here below, we compare this enumerative procedure to our cutting plane algorithm on several instances, and show that the latter one is a lot more effective in practice than the former one. In the next table, Enum2 and Enum3 report the CPU time in seconds of the enumerative algorithm, for $L = 2$ and 3, respectively.

Clearly, for the instances tested, the cutting plane algorithm performed a lot faster than the enumerative procedure. In particular, when $(n, k) = (20, 5)$, $(20, 10)$ or $(40, 5)$, for $L = 3$, while the former one gave the optimal solution in less than one second, the latter one had not yet reached optimality after one day of computations.

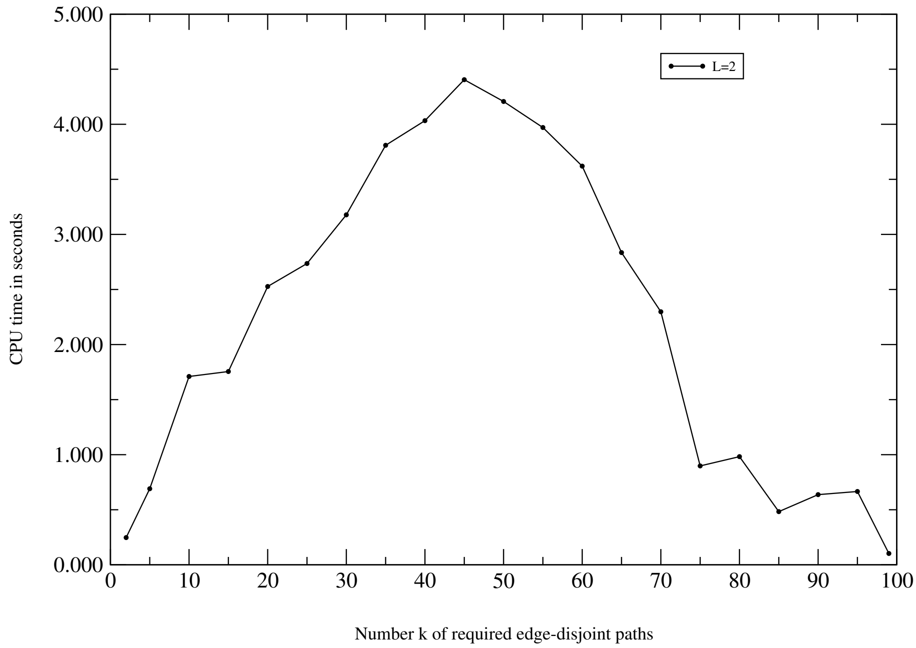


Figure 3.2: CPU time in function of $k$ for a given $k$HPP instance when $L = 2$

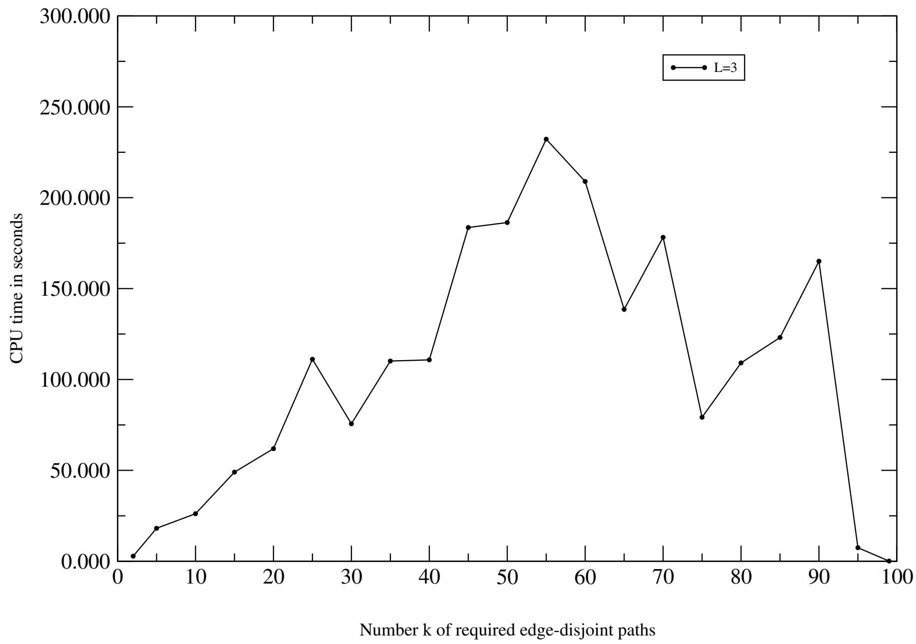


Figure 3.3: CPU time in function of $k$ for a given $k$HPP instance when $L = 3$

Table 3.2: Respective CPU times of the enumerative and cutting plane algoritms for $L = 2, 3$

| $n$ | $k$ | Enum2 | CPU2 | Enum3 | CPU3 |
|---|---|---|---|---|---|
| 20 | 2 | 0.03 | 0.02 | 0.89 | 0.03 |
| 20 | 5 | 0.43 | 0.02 | > 86400.00 | 0.03 |
| 20 | 10 | 6.16 | 0.02 | > 86400.00 | 0.06 |
| 40 | 2 | 0.07 | 0.04 | 81.95 | 0.05 |
| 40 | 5 | 91.15 | 0.07 | > 86400.00 | 0.51 |
| 100 | 2 | 2.78 | 1.52 | 18303.94 | 4.51 |

## 3.5 Concluding remarks

In this chapter, we have given a complete and minimal description of the polytope associated to the problem of finding a minimum cost subgraph with at least $k$ edge-disjoint 2-$st$-paths, for any $k \geq 2$. This have answered part of Conjecture 2.8.1 from Chapter 2.

Moreover, we have presented some computational experiments of a cutting plane algorithm for this problem when $L = 2$ or 3. Thanks to the polynomial separation procedures used in it, we have reached optimality in a few minutes for instances up to 100 nodes and arbitrary values $k$ of required paths. This appeared to be a lot better than the performance of the enumerative procedure, which is only polynomial for $k$ fixed. We have also analyzed the variation of the computing time in function of $k$, and shown that this one is maximum when $k$ is about half the number of nodes in the graph. Finally, all optimal solutions obtained when $L = 3$ were integer, which gives hope for Conjecture 2.8.1 being still true when $L = 3$.

Of course, the $k$HPP can be seen as a special case of the more general problem when more than one pair of terminals is considered. The efficient cutting plane procedure used here for solving the $k$HPP will be our starting point in order to devise a good algorithm for that new problem. Our primary goal will then be to determine new classes of facet defining inequalities for this more general problem. Separation procedures for these classes will be embedded in the previous algorithm. We will also make use of branching this time. The whole Branch-and-Cut algorithm, along with a polyhedral study of the problem, will be presented in the next two chapters.

Finally, it would be interesting to study from a polyhedral point of view the $k$HPP for $L \geq 3$. For $k = 2$ and $L = 3$, we already shown in Chapter 2 that the $st$-cut, $L$-$st$-path-cut and trivial inequalities describe completely $P_k(G, L)$. However, for $k = 2$ and $L = 4$, this system does not suffice to even formulate the $k$HPP as an integer program. This question will be considered in Chapter 6.

# Chapter 4

# Rooted Two-Edge Connected Hop-Constrained Network Design Problem

In this chapter and the next one, we will consider the *Two-edge connected Hop-constrained Network Design Problem* (or THNDP for short). So, if the number of required edge-disjoint $L$-paths will be fixed to 2 again, we will work this time with a non-singleton set $D$ of demands. Nevertheless, for most of the results given in this chapter, we will restrict our attention to $L = 2, 3$, and $D$ being rooted, that is, all demand pairs having one node in common. This work has been the object of an article [45]. In Chapter 5, we will then consider in more depth the case where $D$ is an arbitrary set of demands.

## 4.1 Introduction

Let $G = (N, E)$ be a graph. Let $D \subseteq N \times N$ be a set of pairs of nodes, called *demands*. If the pair $\{s, t\}$ is a demand in $D$, we will call $s$ and $t$ *demand nodes* or *terminal nodes*. In particular, when several demands $\{s, t_1\}, \ldots, \{s, t_d\}$ are rooted in the same node $s$, we will speak of $s$ as a *source node* and of the $t_i$'s as the *destination nodes* of $s$. The nodes in $N$ that do not belong to any demand of $D$, will be called *Steiner nodes*.

Let $L \geq 2$ be a fixed integer. If $s, t$ are two nodes of $N$, recall that an *$L$-st-path* in $G$ is a path between $s$ and $t$ of length at most $L$, where the length of a path is the number of its edges (or *hops*). Given a function $c : E \rightarrow \mathbb{R}$ which associates a cost $c(e)$ to each edge $e \in E$, the *Two-edge connected Hop-constrained Network Design Problem* (THNDP) is to find a minimum cost subgraph such that, between each demand $\{s, t\} \in D$, there exist at least two edge-disjoint $L$-$st$-paths. If all the demands in $D$ are rooted in some node $s$, then we will speak about the *rooted* THNDP.

In this chapter, we study the THNDP from a polyhedral point of view. We first show that the (rooted) THNDP is strongly NP-hard for any fixed value of $L \geq 2$. We then give an integer programming formulation for the THNDP with $L = 2, 3$. We also introduce several classes of valid inequalities, along with necessary conditions and sufficient conditions for these inequalities to be facet-defining in the rooted case. We finally discuss separation routines for these classes of inequalities, and, using them, we propose a Branch-and-Cut algorithm for the THNDP when $L = 2, 3$. We conclude by giving some computational results based on random and real instances, in the rooted and arbitrary cases, for $L = 2$ or 3.

Recall that, for a partition $\Pi = (V_0, V_1, \ldots, V_p)$ of $N$, the associated *multicut* in $G$, denoted by $\Delta_\Pi(G) = \delta(V_0, V_1, \ldots, V_p)$, is the set of edges having their end nodes in two different subsets. We will then denote by $E_\Pi^{q,r} = \bigcup_{i=q,q+1,\ldots,r} [V_i, V_{i+1}]$ the set of edges between the consecutive subsets $V_q, V_{q+1}, \ldots, V_{r+1}$ of $\Pi$. Also, given a set of demands $D \subseteq N \times N$ and an integer $L \geq 2$, the convex hull of the incidence vectors of the solutions to the THNDP on $G$, denoted by $P_G(D, L)$, will be called the THNDP *polytope*.

The chapter is organized as follows. In the next section, we investigate the complexity of the (rooted) THNDP. In Section 4.3, we give an integer programming formulation for the problem when $L = 2, 3$. In Section 4.4, we present some new classes of valid inequalities. Necessary and sufficient conditions for these inequalities to be facet-defining are discussed in Section 4.5. In Section 4.6, we study the separation of these inequalities. In Section 4.7, we derive a Branch-and-Cut algorithm and present our experimental results. Finally, in Section 4.8, we give some concluding remarks.

## 4.2 Complexity of the THNDP

It is easy to see that the rooted THNDP is a generalization of the two-edge connected subgraph problem. This consists, given a graph with weights on its edges, in finding a minimum weight subgraph such that, between every pair of nodes, there exist at least two edge-disjoint paths. So the two-edge connected subgraph problem is nothing but the rooted THNDP for $D = \{s\} \times N$ and $L$ sufficiently large. It is well known that this former problem is $NP$-hard, even when the weights are all equal to 1. This implies that the rooted THNDP is also $NP$-hard in this case. Moreover, since the input is then of polynomial length, we have that the rooted THNDP is strongly $NP$-hard.

In what follows, we are going to show that the rooted THNDP remains strongly $NP$-hard for every $L \geq 2$. Of course, since the rooted THNDP is a particular case of the THNDP, the same will hold for the latter problem.

**Theorem 4.2.1.** *For $L \geq 2$ fixed, if $P \neq NP$, then the rooted THNDP is strongly $NP$-hard.*

*Proof.* We will show that any instance of the minimum cardinality dominating set problem can be polynomially transformed to an instance of the rooted THNDP with $L \geq 2$. As the former problem is $NP$-hard (see [27]), this will prove that the latter one is also $NP$-hard. Moreover, as the input data of the corresponding THNDP instance will always be of polynomial length, we obtain that this problem is strongly $NP$-hard.

Consider an instance $G = (N, E)$ of the dominating set problem. This problem consists in finding a subset $N'$ of $N$ of minimal cardinality such that every node in $N \backslash N'$ is adjacent to at least one node of $N'$.

Let us construct an instance of the rooted THNDP for a fixed value of $L$ in the following way. We create a source node $s$, and two copies, $N_1$ and $N_2$, of $N$. Consider an edge $sv$ for each node $v \in N_1 \cup N_2$, and an edge $v_1v_2$ between $v_1 \in N_1$ and $v_2 \in N_2$ if the corresponding nodes in $N$ are either the same or adjacent to each other in the original graph $G$. Finally, insert $L-2$ nodes of degree two on each edge between either $s$ and $N_2$, or $N_1$ and $N_2$. Observe that the latter operation transforms these edges into paths of length $L-1$, which we denote in the following way. For each $u \in \{s\} \cup N_1$ and $v \in N_2$, let us call $P_{uv}$ the $(L-1)$-$uv$-path. Let us denote by $\overline{G} = (\overline{N}, \overline{E})$ the auxiliary graph. See Fig. 4.1 for an illustration for $L = 3$. We consider the rooted THNDP on $\overline{G}$, with unit costs on all edges, and the set $N_2$ as destination nodes relatively to $s$.

Let $S^*$ be an optimal solution to the rooted THNDP in $\overline{G}$ with respect to $s$ and $N_2$. In what follows, we are going to show that an optimal solution of the THNDP in $\overline{G}$ corresponds to a minimum cardinality dominating set in $G$ and conversely. Let $\Gamma_1 = \bigcup_{v \in N_2} P_{sv}$ and $\Gamma_2 = \bigcup_{u \in N_1, v \in N_2} P_{uv}$.

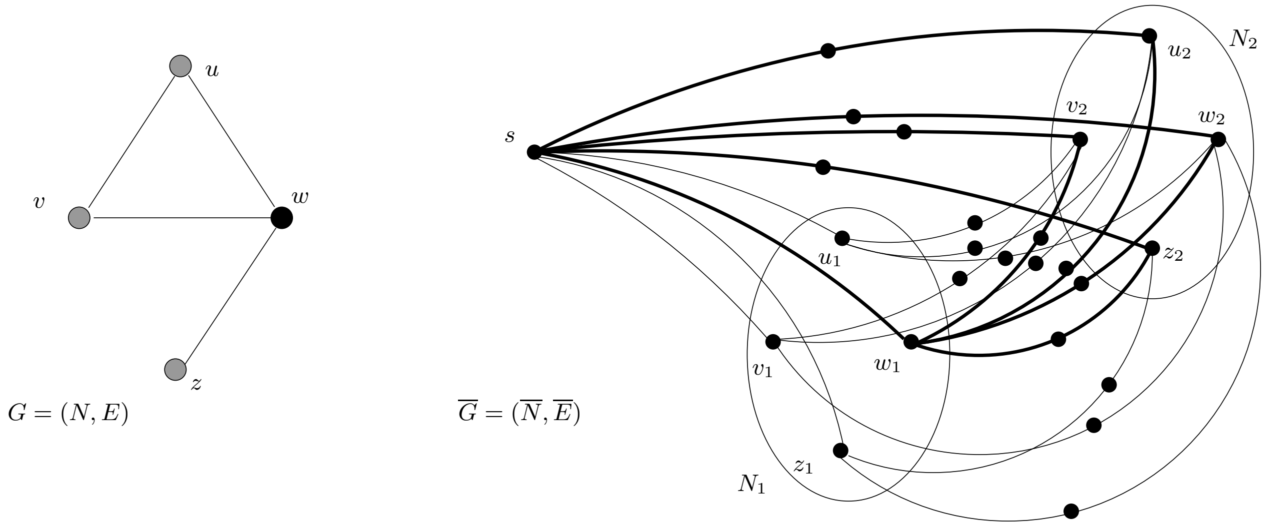


Figure 4.1: An instance of the dominating set problem (graph $G$) and the corresponding instance of the rooted THNDP with $L = 3$ (graph $\overline{G}$)

*Claim 1.* There exists an optimal solution $S^*$ of the rooted THNDP in $\overline{G}$ such that

(i) $S^*$ contains all the paths of $\Gamma_1$, and

(ii) $S^*$ contains exactly $|N|$ paths from $\Gamma_2$.

*Proof.* First remark that any feasible solution to the THNDP will contain at least $2|N|$ paths from $\Gamma_1 \cup \Gamma_2$, since such a path used for an $L$-$sv_2$-path, with $v_2 \in N_2$, cannot be used for an $L$-$sv_2'$-path, where $v_2' \in N_2 \backslash \{v_2\}$.
Moreover, an optimal solution $S^*$ for the THNDP in $\overline{G}$ can be considered so that it contains the $|N|$ paths of $\Gamma_1$. Indeed, if a path of $\Gamma_1$, say $P_{sv_2}$ with $v_2 \in N_2$, is not taken in $S^*$, then there must exist two $L$-$sv_2$-paths in the solution, going through $N_1$.

Therefore, the optimal solution contains two paths from $\Gamma_2$ incident to $v_2$. Now, by replacing one of these two by $P_{sv_2}$, we get a solution with the same cardinality, hence being still optimal, and containing $P_{sv_2}$. This shows (i).
Finally, by (i) and the first remark, we have that $S^*$ must contain at least $|N|$ paths in $\Gamma_2$. As, for each node $v_2 \in N_2$, only one path from $\Gamma_2$ is needed in $S^*$, and since the costs are positive, it follows that $S^*$ contains exactly $|N|$ paths from $\Gamma_2$, which establishes (ii). ♦

By Claim 1, all the paths of $\Gamma_1$ can be considered in $S^*$ and $S^*$ then contains exactly $|N|$ paths from $\Gamma_2$. As a consequence, determining such an optimal solution $S^*$ reduces to finding a minimum number of edges between $s$ and $N_1$ such that, for each node $v_2 \in N_2$, there is a path of length exactly $L$ between $s$ and $v_2$, going through $N_1$ and using one of those $|N|$ paths of $\Gamma_2$. Since every node of $N_1$ is adjacent to $s$, this is equivalent to finding a minimum cardinality subset $N_1'$ of $N_1$ that *covers* all the nodes of $N_2$, in the sense that all the nodes of $N_2$ are reachable by the paths of $\Gamma_2$ going out of $N_1'$. See for instance Fig. 4.1 where $S^*$ is represented by the bold edges.

*Claim 2.* A node subset $N'$ of $N$ is a dominating set of $G$ if and only if the corresponding subset $N_1'$ of $N_1$ covers all the nodes of $N_2$ in $\overline{G}$.

*Proof.* Suppose that we have a dominating set $N'$ of $G$. By definition, each node of $N \backslash N'$ is adjacent to at least one node of $N'$. Let $N_1'$ and $N_2'$ be the sets of nodes corresponding to $N'$ in $N_1$ and $N_2$, respectively. In $\overline{G}$, we have that each node in $N_2 \backslash N_2'$ is adjacent to at least one node of $N_1' \subseteq N_1$ by a path of $\Gamma_2$. Moreover, by construction, each node of $N_2'$ is adjacent to its copy in $N_1'$. Hence, we have that $N_2$ is covered by the paths of $\Gamma_2$ going out of $N_1'$. The other way is similar. ♦

By Claim 2, and the above developments, the minimum cardinality subset $N_1' \subseteq N_1$ that covers $N_2$ corresponds to a minimum cardinality dominating set in $G$ (for example, the black node $w$ in Figure 4.1). Moreover, it is clear that any optimal solution in $\overline{G}$ can be transformed into a solution verifying the conditions of Claim 1 in polynomial time. □

For rooted demands and unitary costs, along with the additional assumption that the underlying graph is complete, the THNDP can be solved in polynomial time, as shown in the following theorem. This result is based on Theorem 4.4.5 whose proof in given in Section 4.4.

**Theorem 4.2.2.** *If the graph $G$ is complete, and all edge costs are equal to 1, the rooted THNDP can be solved in polynomial time for every $L \geq 2$.*

*Proof.* Let $D = \{\{s,t_1\},\{s,t_2\},\ldots,\{s,t_d\}\}$ be a set of $d$ demands rooted in some node $s$. By Theorem 4.4.5, the minimum number of edges of a feasible solution to the THNDP is $\lceil (L+1)d/L \rceil$. Since the graph $G$ is complete, it is easy to build a solution having exactly this number of edges, hence optimal for the THNDP in $G$ with unitary costs. This solution can be constructed in the following way. We consider the first $L$ destination nodes and cover them, along with $s$, with a simple cycle of length $L+1$. Observe that each destination node $t_i$ that is considered is covered by two edge-disjoint $L$-$st_i$-paths. This procedure is iterated for sets of $L$ destination nodes until there remain $l < L$ destination nodes to cover. If $l = 0$, we are done. If $l = 1$, we link the last destination node $t_d$ to $s$ and to $t_{d-1}$ (two additional edges). The resulting solution contains $\lceil (L+1)d/L \rceil$ edges. If $l > 1$, we cover the $l$ remaining destination nodes, along with $s$, by a simple cycle of length $l+1$, and the obtained solution still contains the desired number of edges. □

## 4.3 Integer programming formulation

It is clear that the incidence vector $x^F$ of any solution $(N,F)$ to the THNDP for any $L \geq 2$ satisfies the following inequalities.

$$x(\delta(W)) \geq 2, \quad \text{for all } st\text{-cut } \delta(W), \text{ for all } \{s,t\} \in D, \tag{4.1}$$

$$x(T) \geq 2, \quad \text{for all } L\text{-}st\text{-path-cut } T, \text{ for all } \{s,t\} \in D, \tag{4.2}$$

$$x(e) \leq 1, \quad \text{for all } e \in E, \tag{4.3}$$

$$x(e) \geq 0, \quad \text{for all } e \in E. \tag{4.4}$$

Recall that inequalities (4.1), (4.2), and (4.3)-(4.4) are respectively called *st-cut inequalities*, *L-(st-)path-cut inequalities*, and *trivial inequalities*. Also, with the notations of Section 4.1, note that an $L$-path-cut $T$ is equal to $\Delta_\Pi(G)\backslash E_\Pi^{0,L}$.

It is clear that the system of inequalities (4.1)-(4.4) along with integrality constraints formulates the THNDP as an integer program when $L = 2,3$. The proof being similar to the one for a single demand $\{s,t\}$ (see Theorem 2.2.3), we only give the theorem and its lemma.

**Lemma 4.3.1.** *Let $G=(N,E)$ be a graph, $s$ and $t$ two nodes of $N$, and $L\in\{2,3\}$. Suppose that there do not exist two edge-disjoint $L$-st-paths in $G$. If $G$ contains an $L$-st-path, then there exists an edge that belongs to every $L$-st-path.*

**Theorem 4.3.2.** *Let $G=(N,E)$ be a graph and $L\in\{2,3\}$. Then the THNDP is equivalent to the integer program*

$$\textit{Min}\ \{cx : x \textit{ satisfies (4.1)-(4.4)}, x\in\mathbb{Z}^E\}.$$

Of course, this result is no more true when $L\geq 4$, since this was already the case when $|D|=1$ (see Chapter 2). The problem seems to be much harder in that case. We have however identified other classes of valid inequalities for the THNDP, for any $L\geq 2$. The next section is devoted to these classes.

## 4.4 Valid inequalities

In what follows, we present new classes of valid inequalities for the THNDP for any $L\geq 2$ and set of demands $D=\{\{s_1,t_1\},\ldots,\{s_d,t_d\}\}$ with $d\geq 2$. Note that some demands may have common nodes.

**Theorem 4.4.1.** *Let $G=(N,E)$ and $\Pi=(V_0,V_1,\ldots,V_{L+1})$ be a partition of $N$ with $s_1\in V_0$ (resp. $t_1\in V_0$) and $t_1\in V_{L+1}$ (resp. $s_1\in V_{L+1}$). Let $i\in\{0,\ldots,L\}$ such that $V_i$ and $V_{i+1}$ induce st-cuts. Let*
*$\overline{E}=[V_{i-1},V_i]\cup[V_{i+1},V_{i+2}]\cup(\bigcup_{k,l\notin\{i,i+1\},\,|k-l|>1}[V_k,V_l])$,*
*and $F\subseteq\overline{E}$. Then the inequality*

$$x(\Delta_\Pi(G)\backslash(F\cup E_\Pi^{0,i-2}\cup E_\Pi^{i+2,L}))\geq 3-\lfloor|F|/2\rfloor \tag{4.5}$$

*is valid for $P_G(D,L)$.*

*Proof.* Observe that the following inequalities are valid for the THNDP.

$$\begin{array}{lll} x(\delta(V_i)) & \geq & 2, \\ x(\delta(V_{i+1})) & \geq & 2, \\ x(\Delta_\Pi(G)\backslash E_\Pi^{0,L}) & \geq & 2, \\ -x(f) & \geq & -1, \quad \text{for all } f \in F, \\ x(f) & \geq & 0, \quad \text{for all } f \in \overline{E}\backslash F. \end{array}$$

The first two inequalities indeed correspond to the $st$-cuts induced by $V_i$ and $V_{i+1}$, respectively, and the third one is the $L$-$s_1t_1$-path-cut inequality induced by $\Pi$. By summing these inequalities, dividing the sum by 2, and rounding up the righthand side to the highest integer, we obtain inequality (4.5). □

Inequality (4.5) will be called *double cut inequality*, and the set of edges having a positive coefficient in (4.5) *double cut.*

In particular, these inequalities will take the following form when $L = 2, 3$, $|F| = 1$ and $i = 0$ (in that case $V_{i-1}$ does not exist).

**Corollary 4.4.2.** *i) Let $L = 2$ and $(V_0, V_1, V_2, V_3)$ be a partition of $N$ such that $s_1 \in V_0$ (resp. $t_1 \in V_0$), $t_2 \in V_1$ and $t_1 \in V_3$ (resp. $s_1 \in V_3$). Let $e \in [V_1, V_2]$. The double cut inequality*

$$x(\delta(V_0)) + x([V_1, V_3]) + x([V_1, V_2]\backslash\{e\}) \geq 3 \tag{4.6}$$

*is valid for $P_G(D, 2)$.*

*ii) Let $L = 3$ and $(V_0, \ldots, V_4)$ be a partition of $N$ such that $s_1 \in V_0$ (resp. $t_1 \in V_0$), $t_2 \in V_1$ and $t_1 \in V_4$ (resp. $s_1 \in V_4$). Let $e \in [V_1, V_2] \cup [V_2, V_4]$. The double cut inequality (see Fig. 4.2)*

$$x(\delta(V_0)) + x([V_1, V_3 \cup V_4]) + x([V_1 \cup V_4, V_2]\backslash\{e\}) \geq 3 \tag{4.7}$$

*is valid for $P_G(D, 3)$.*

Our second class of valid inequalities are called *triple path-cut inequalities.* In a similar way as before, we will speak of a *triple path-cut* as the set of the edges having a positive coefficient in such an inequality. They are defined as follows.

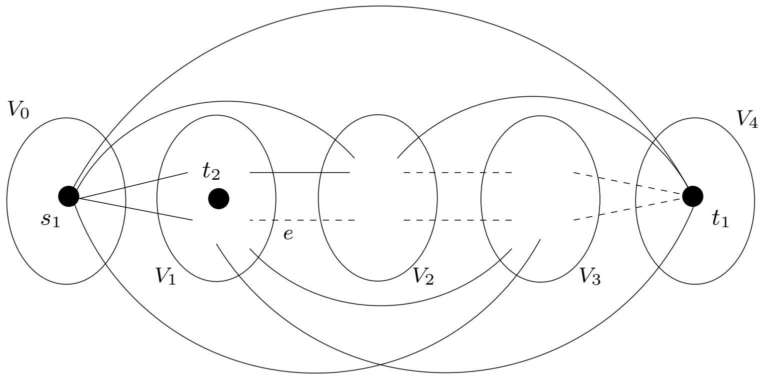


Figure 4.2: Support graph of a double cut inequality for $L = 3$

**Theorem 4.4.3.** *Let $L \geq 4$ and $\Pi = (V_0, V_1, \dots, V_{L+2})$ be a partition of $N$ such that $s_1, s_2 \in V_0$, $t_1 \in V_{L+1}$, and $t_2 \in V_{L+2}$. Let $e \in [V_L, V_{L+1} \cup V_{L+2}]$. Then the inequality*

$$2x(\Delta_\Pi(G)\backslash E_\Pi^{0,L-1}) - x(\textstyle\bigcup_{j=0,1,2,L}[V_j, V_{L+1} \cup V_{L+2}]) - x([V_{L-1} \cup V_{L+2}, V_{L+1}]) - x(e) \geq 3, \tag{4.8}$$

*is valid for $P_G(D, L)$.*

*Proof.* Let $T_1$ be the $L$-$s_1t_1$-path-cut induced by the partition $(V_0, V_1 \cup V_{L+2}, V_2, V_3, \dots, V_{L-1}, V_L, V_{L+1})$, and $T_2$ and $T_3$ be the $L$-$s_2t_2$-path-cuts induced by $(V_0, V_1 \cup V_{L+1}, V_2, V_3, \dots, V_{L-1}, V_L, V_{L+2})$ and $(V_0, V_1, V_2, V_3, \dots, V_{L-1}, V_L \cup V_{L+1}, V_{L+2})$, respectively. Then the following inequalities are valid for the THNDP.

$$\begin{array}{rcl} x(T_1) & \geq & 2, \\ x(T_2) & \geq & 2, \\ x(T_3) & \geq & 2, \\ -x(e) & \geq & -1. \end{array}$$

By summing these inequalities, along with nonnegativity constraints, dividing the sum by 2, and rounding up the righthand side, we obtain inequality (4.8). □

When $L = 2, 3$, triple path-cut inequalities have a slightly different form, due to multiple occurences of certain sets of edges in the lefthand side of (4.8). The way of obtaining them stays however the same.

**Corollary 4.4.4.** *i) Let $L = 2$ and $(V_0, V_1, \dots, V_4)$ be a partition of $N$ with $s_1, s_2 \in V_0$, $t_1 \in V_3$ and $t_2 \in V_4$. Then the triple path-cut inequality (see Fig. 4.3)*

$$2x([V_0, V_2]) + x([V_0, V_3 \cup V_4]) + x([V_1, V_3]\backslash\{e\}) + x([V_1 \cup V_3, V_4]) \geq 3, \tag{4.9}$$

*where* $e \in [V_1, V_3]$*, is valid for* $P_G(D, 2)$.

*ii) Let* $L = 3$ *and* $(V_0, V_1, \ldots, V_5)$ *be a partition of* $N$ *with* $s_1, s_2 \in V_0$*,* $t_1 \in V_4$ *and* $t_2 \in V_5$*. Then the triple path-cut inequality*

$$2x([V_0, V_2 \cup V_3]) + 2x([V_1, V_3]) + x([V_0 \cup V_1 \cup V_2 \cup V_3, V_4 \cup V_5] \setminus \{e\}) + x([V_4, V_5]) \geq 3, \quad (4.10)$$

*where* $e \in [V_2 \cup V_3, V_4] \cup [V_3, V_5]$*, is valid for* $P_G(D, 3)$.

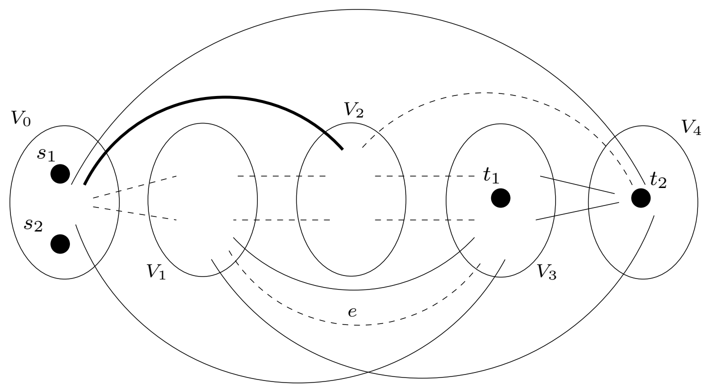


Figure 4.3: Support graph of a triple path-cut inequality for $L = 2$

Our next class of inequalities gives the minimum number of edges used by a feasible solution of the THNDP in a multicut, based on a subset of demands rooted in some node $s$. We will call such a multicut a *rooted-partition* and the associated inequality a *rooted-partition inequality.*

**Theorem 4.4.5.** *Let* $L \geq 2$ *and* $T = \{t_1, \ldots, t_p\}$ *be a subset of* $p$ *destination nodes relatively to node* $s$*. Let* $\Pi = (V_0, V_1, \ldots, V_p)$ *be a partition of* $N$ *such that* $s \in V_0$*, and* $t_i \in V_i$*, for all* $i = 1, \ldots, p$*. See Fig. 4.4 for an illustration. Then the inequality*

$$x(\Delta_\Pi(G)) \geq p + \lceil p/L \rceil \; (= \lceil (L+1)p/L \rceil) \quad (4.11)$$

*is valid for* $P_G(D, L)$.

*Proof.* The proof is by induction on $p$. If $p = 1$, it is obvious that we need at least two edges in any feasible solution, and thus (4.11) is satisfied. Suppose that the statement holds for any partition based on at most $p - 1$ destination nodes relatively to $s$. We will show that the statement remains true for the partition $\Pi$.

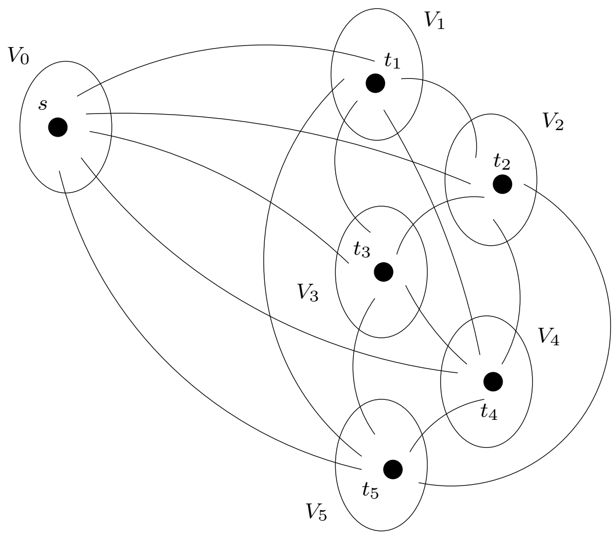


Figure 4.4: Support graph of a rooted-partition inequality on $p = 5$ demands

First remark that we can assume that the $V_i$'s contain only one node, that is $V_0 = \{s\}$ and $V_i = \{t_i\}$ for $i = 1, \ldots, p$. If not, we could indeed consider the graph obtained from $G$ by contracting the $V_i$'s with $|V_i| \geq 2$. It is clear that neither the number of edges in the multicut, nor the number $p$ of distinguished destinations, would be changed by this operation, and hence the inequality to prove remains the same.
Now let $F$ be a feasible solution of the THNDP. Let $T_1$ (resp. $T_2$) be the subset of destinations adjacent to $s$ (resp. not adjacent to $s$) in $F$. By the previous remark, we can assume that $T_1 \neq \emptyset$. Let $t_1 \in T_1$. There must exist in $F$ another $L$-path between $s$ and $t_1$. Note that this path can use an edge that is parallel to the first edge between $s$ and $t_1$. These two paths give us a cycle $C$ of length $c$, with $2 \leq c \leq L+1$, going through at most $c-1$ destinations. Suppose, w.l.o.g., that $C = \{s, t_1, \ldots, t_{c-1}\}$. Let $\Pi' = (V'_0, V'_1, \ldots, V'_{p-c+1})$ be the partition given by $V'_0 = \{s, t_1, \ldots, t_{c-1}\}$ and $V'_i = V_{i+c-1}$ for $i = 1, \ldots, p-c+1$. By induction, we have

$$x(\Delta_{\Pi'}(G)) \geq (p-(c-1)) + \lceil (p-(c-1))/L \rceil.$$

This yields

$$x(\Delta_{\Pi}(G)) \geq (p-(c-1)) + \lceil (p-(c-1))/L \rceil + c.$$

Since $c \leq L+1$, we have that $p-(c-1) \geq p-L$. Hence $\lceil (p-(c-1))/L \rceil \geq \lceil (p-L)/L \rceil = \lceil p/L \rceil - 1$. So,

$$\begin{aligned} x(\Delta_{\Pi}(G)) &\geq p + 1 + \lceil p/L \rceil - 1 \\ &= p + \lceil p/L \rceil. \end{aligned}$$

□

We will now present a reinforcement of the rooted-partition inequalities when $L = 2$.

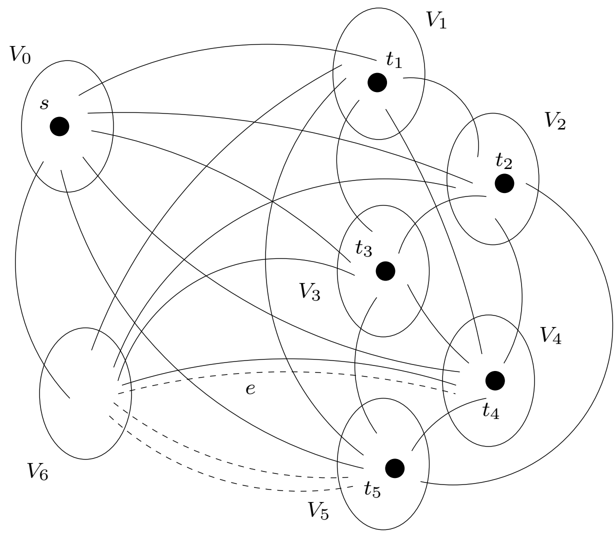


Figure 4.5: Support graph of a rooted-opt-partition inequality on $p = 5$ demands

**Theorem 4.4.6.** *Let $L = 2$ and $T = \{t_1, \ldots, t_p\}$ be a subset of $p \geq 2$ destination nodes of $s$ amongst the demands of $D$. Let $\Pi = (V_0, \ldots, V_{p-1}, V_p, V_{p+1})$ be a partition of $N$ such that $s \in V_0$, and $t_i \in V_i$, for all $i = 1, \ldots, p$. Let $e \in [V_{p-1}, V_{p+1}]$. See Fig. 4.5 for an illustration. Then the following inequality*

$$x(\Delta_\Pi(G) \backslash ([V_p, V_{p+1}] \cup \{e\})) \geq \lceil 3p/2 \rceil, \tag{4.12}$$

*is valid for $P_G(D, 2)$.*

*Proof.* If $p = 2$, it is obvious that we need at least three edges in any feasible solution, and thus (4.12) is satisfied. Suppose that the statement holds for any partition based on at most $p - 1$ destination nodes relatively to $s$. We will show by induction that the statement remains true for the partition $\Pi$.

First note that, as we did in Theorem 4.4.5, we can suppose that the sets $V_i$, $i = 0, \ldots, p+1$, are reduced to single nodes. Also we may suppose that the (single) node of $V_{p+1}$, say $v$, is a Steiner node. Indeed, if inequality (4.12) is valid in this case, it would also be valid if the node of $V_{p+1}$ is a terminal node.
Consider the rooted-partition inequality induced by the partition $(V_0, \ldots, V_{p-1} \cup V_{p+1}, V_p)$,

$$x(\delta(V_0, \ldots, V_{p-1} \cup V_{p+1}, V_p)) \geq \lceil 3p/2 \rceil.$$

Similarly, by collapsing $V_p$ and $V_{p+1}$, we get

$$x(\delta(V_0, \ldots, V_{p-1}, V_p \cup V_{p+1})) \geq \lceil 3p/2 \rceil.$$

Since $\Delta_\Pi(G) = \delta(V_0, \ldots, V_{p-1} \cup V_{p+1}, V_p) \cup [V_{p-1}, V_{p+1}] = \delta(V_0, \ldots, V_{p-1}, V_p \cup V_{p+1}) \cup [V_p, V_{p+1}]$, the previous two inequalities can be respectively written as

$$x(\Delta_\Pi(G)) - x([V_{p-1}, V_{p+1}]) \geq \lceil 3p/2 \rceil, \tag{4.13}$$

and

$$x(\Delta_\Pi(G)) - x([V_p, V_{p+1}]) \geq \lceil 3p/2 \rceil. \tag{4.14}$$

Let $F$ be a solution to the THNDP. We distinguish two cases.

**Case 1:** $e \notin F$ or $F \cap [V_p, V_{p+1}] = \emptyset$. If $e \notin F$, then, by (4.14), we have that (4.12) is satisfied by $x^F$, the incidence vector of $F$. If $F$ does not intersect $[V_p, V_{p+1}]$, then inequality (4.13) implies (4.12), and thus this latter inequality is satisfied by $x^F$.

**Case 2:** $e \in F$ and $F \cap [V_p, V_{p+1}] \neq \emptyset$. First note that we may suppose that $F \cap [s, V_{p+1}] \neq \emptyset$, because otherwise $F \backslash \{e\}$ would be a solution of the THNDP, and Case 1 would apply. So let us consider an edge, say $e_0$, of $F \cap [s, V_{p+1}]$. Let also $e'$ be an edge in $F \cap [V_p, V_{p+1}]$.
Suppose that there is a cycle $C$ of length $c = 2$ or 3, linking $s$ and $c-1$ destinations from $\{t_1, \ldots, t_{p-2}\}$. W.l.o.g., we may suppose that $C$ is simple and $C = \{s, t_1, \ldots, t_{c-1}\}$. Let $\Pi' = (V'_0, \ldots, V'_{p-c+2})$ such that $V'_0 = \bigcup_{i=0,\ldots,c-1} V_i$, and $V'_i = V_{i+c-1}$, for $i = 1, \ldots, p-c+2$. We have

$$x^F(\Delta_{\Pi'}(G)) = x^F(\Delta_\Pi(G)) - c. \tag{4.15}$$

By induction hypothesis, inequality (4.12) is satisfied by $x^F$ with respect to $\Pi'$ and $e$, and hence we get

$$x^F(\Delta_{\Pi'}(G) \backslash ([V'_{p-c+1}, V'_{p-c+2}] \cup \{e\})) \geq \lceil 3(p-c+1)/2 \rceil.$$

As $c \leq 3$, by (4.15), it follows that inequality (4.12) is also satisfied by $x^F$ with respect to $\Pi$ and $e$. Thus we may suppose that the subgraph, say $\tilde{G} = (\tilde{V}, \tilde{E})$, induced by $F$ on the nodes $s, t_1, \ldots, t_{p-2}$ does not contain any cycle of length 2 or 3.
Let $T_1$ (resp. $T_2$) be the set of nodes among $\{t_1, \ldots, t_{p-2}\}$ that are linked (resp. not linked) to $s$ by edges in $F$. Note that, for every node $t_i \in T_1$, we already have one 2-$st_i$-path, namely the edge $st_i$. Since $\tilde{G}$ does not contain cycles of length 2 or 3, the second 2-$st_i$-path for $t_i \in T_1$ must use one edge of $[t_i, \{t_{p-1}, t_p, v\}]$. Moreover, all those edges are different. Therefore, to cover $T_1$, $F$ uses at least $2|T_1|$ edges.

Next, each node in $T_2$ must be linked to $s$ by two 2-paths, each one going through either $T_1$ or $\{t_{p-1}, t_p, v\}$. Hence $F$ must use two additional edges for every node of $T_2$. In consequence, in the 2-paths between $s$ and the nodes $\{t_1, \dots, t_{p-2}\}$, $F$ uses at least $2(|T_1| + |T_2|) = 2(p-2)$ edges.
Let $\tilde{F}$ be the set of these edges, and $\tilde{F}'$ be the set of edges of $F$ different from those in $\tilde{F} \cup [V_p, V_{p+1}] \cup \{e\}$. We have the following claim.

*Claim.* $|\tilde{F}'| \geq 3$.

*Proof.* First note that $e_0 \in \tilde{F}'$. Also observe that, in order to cover $t_{p-1}, t_p$, besides $e_0, e, e'$, we still need one 2-path from $s$ to $t_{p-1}$ and another from $s$ to $t_p$. (Recall that $v$ is a Steiner node.) We consider two cases.
**Case 2.1:** None of the edges of $\tilde{F}$ is incident to $t_{p-1}$ or $t_p$. This implies that the edges of $\tilde{F}$ used in 2-$st_i$-paths, with $t_i \in T_1 \cup T_2$, going through the node set $\{t_{p-1}, t_p, v\}$ are all incident to $v$. So, to cover $t_{p-1}$, we need at least one more edge, say $h$, from $\delta(t_{p-1}) \backslash \{e\}$. Moreover, we note that $h \in \tilde{F}'$. If $h \notin [t_{p-1}, t_p]$, then, in order to cover $t_p$, one more edge in $\tilde{F}'$ is needed, and hence, $|\tilde{F}'| \geq 3$. If $h \in [t_{p-1}, t_p]$, again it is not hard to see that one more edge from $\tilde{F}'$ is necessary to cover both $t_{p-1}$ and $t_p$.
**Case 2.2:** Some of the edges of $\tilde{F}$ are incident to $t_{p-1}$ or $t_p$. If there is an edge of $\tilde{F}$ incident to $t_{p-1}$ (resp. $t_p$), then $\tilde{F}'$ must contain the edge $st_{p-1}$ (resp. $st_p$). If not, the edge of $\tilde{F}$ would not belong to a 2-$st_i$-path, where $t_i \in T_1 \cup T_2$, which is a contradiction. Therefore, if there are edges of $\tilde{F}$ incident to both $t_{p-1}$ and $t_p$, then clearly $\tilde{F}'$ contains at least the edges $e_0, st_{p-1}, st_p$ and thus $|\tilde{F}'| \geq 3$. Now suppose there is only one node $t_i$ among $t_{p-1}, t_p$ that is incident to some edge of $\tilde{F}$. Thus $st_i \in \tilde{F}'$. Moreover, to cover $t_j$, $j \in \{p-1, p\} \backslash \{i\}$, we need one more edge in $\tilde{F}'$, which implies that $|\tilde{F}'| \geq 3$. ♦

In consequence, by the claim above, and as $\tilde{F} \cup \tilde{F}' \subseteq F$, we have that

$$|F| \geq 2(p-2) + 3 \geq \lceil 3p/2 \rceil,$$

for all $p \geq 2$. □

Inequalities (4.12) will be called *rooted-opt-partition inequalities*. The *rooted-opt-partition* will then be the set of edges with positive coefficient in the corresponding inequality. Note that this extension of the rooted-partition inequalities for other values of $L$ leads to non valid inequalities for the THNDP. Indeed, consider a rooted instance where $s$ is the source node, $t_1$ and $t_2$ are the destination nodes of $s$, and $v$ is a Steiner node. A cycle of length four spanning $s, t_1, v, t_2$ in this order is feasible for $L = 3$. However, if we take $V_0 = \{s\}$, $V_1 = \{t_1\}$, $V_2 = \{t_2\}$ and $V_3 = \{v\}$ (hence $p = 2$), the corresponding rooted-opt-partition inequality would be

$$x_{st_1} + x_{st_2} + x_{sv} + x_{t_1t_2} \geq \lceil 4p/3 \rceil = 3,$$

which is violated by this point.

## 4.5 Facets of the rooted THNDP polytope

In this section, we will consider the rooted case, where all the demands are rooted in an unique source node $s$. So, let $G = (N, E)$ be a graph and $D = \{\{s, t_1\}, \ldots, \{s, t_d\}\}$ be a set of rooted demands. We will describe necessary conditions and sufficient conditions for the previous inequalities to be facet-defining. Besides their theoretical interest, these conditions will be used in the next section in order to devise efficient separation procedures.

First we discuss the dimension of $P_G(D, L)$. If $D = \{\{s, t\}\}$, $dim(P_G(D, L)) = |E| - |E^*_{st}|$ where $E^*_{st}$ is the set of $L$-$st$-essential edges of $G$, see Section 2.3. Recall that an edge $e$ is $L$-$st$-essential if and only if $e$ belongs to an $st$-cut or $L$-$st$-path-cut of cardinality 2. If we extend this definition of $E^*_{st}$ to any demand $\{s, t\} \in D$, we get that $dim(P_G(D, L)) = |E| - |\bigcup_{\{s,t\} \in D} E^*_{st}|$. In the following, we will always suppose that $G = (N, E)$ is a complete graph with $|N|$ large enough to have $\bigcup_{\{s,t\} \in D} E^*_{st} = \emptyset$, and hence $P_G(D, L)$ full dimensional.

Let

$$S(G) = \{F \subseteq E \,|\, (N, F) \text{ is a solution of the THNDP}\}.$$

Given an inequality $ax \geq \alpha$ valid for $P_G(D, L)$, we will denote by

$$S_a(G) = \{F \in S(G) \,|\, ax^F = \alpha\}.$$

We first give two lemmas that will be frequently used in the sequel, sometimes without explicit reference. The proofs of similar results having already been given in Section 2.3, they are omitted here.

**Lemma 4.5.1.** *Let $ax \geq \alpha$ be a facet-defining inequality of $P_G(D, L)$, different from the trivial inequalities. Then for every edge $e \in E$, there exists an edge subset in $S_a(G)$ that contains $e$ and another one that does not.*

**Lemma 4.5.2.** *Let $ax \geq \alpha$ be a facet-defining inequality of $P_G(D, L)$, different from a trivial inequality. Then $a(e) \geq 0$, for all $e \in E$ and $\alpha > 0$.*

The following two theorems give conditions for inequalities (4.11) to be facet-defining.

**Theorem 4.5.3.** *Let $G = (N, E)$ be a complete graph and let $t_1, \dots, t_p$ be $p$ destination nodes associated to $s$. Let $\Pi = (V_0, V_1, \dots, V_p)$ be a partition of $N$ such that $s \in V_0$ and $t_i \in V_i$ for $i = 1, \dots, p$. If the rooted-partition inequality (4.11) is facet-defining, then $p$ is not a multiple of $L$.*

*Proof.* Suppose that $p$ is a multiple of $L$. We will show that, for any feasible solution $F$ whose incidence vector $x^F$ satisfies (4.11) with equality, we have $|F \cap [s, T]| = 2p/L$, where $T = \{t_1, \dots, t_p\}$. But this will imply that every solution of the face defined by (4.11) satisfies the equation

$$x([s, T]) = 2p/L.$$

As (4.11) is not a positive multiple of this equation, it cannot define a facet.
For this, first note that, as we did before, we may suppose that the $V_i$'s are reduced to single nodes, that is $V_0 = \{s\}$, $V_i = \{t_i\}$, $i = 1, \dots, p$. It is easy to see that if the statement holds in this case, it also holds when the elements of the partition are not necessarily singletons.

*Claim 1.* The solution $F$ does not contain any chordless cycle containing $s$ of length $\leq L$.

*Proof.* Assume on the contrary that there exists in $F$ a chordless cycle $C$ spanning $s$ and at most $L - 1$ destination nodes of $s$. Suppose, w.l.o.g., that $C$ spans $V_0, V_1, \dots, V_{c-1}$, $c \leq L$. Let $\Pi' = (V'_0, V'_1, \dots, V'_{p-c+1})$ be the partition given by $V'_0 = \bigcup_{i=0,\dots,c-1} V_i$, and $V'_i = V_{i+c-1}$, for $i = 1, \dots, p - c + 1$. By the validity of the rooted-partition inequality induced by $\Pi'$, we have

$$|\Delta_{\Pi'}(G) \cap F| \geq \lceil (L + 1)(p - c + 1)/L \rceil.$$

On the other hand, we have $|\Delta_{\Pi}(G) \cap F| = (L + 1)p/L$. As $|\Delta_{\Pi}(G) \cap F| = |\Delta_{\Pi'}(G) \cap F| + c$, it follows that

$$(L + 1)p/L \geq \lceil (L + 1)(p - c + 1)/L \rceil + c = \lceil (p - c + 1)/L \rceil + p + 1.$$

As $c \leq L$, we have

$$p/L \geq \lceil (p - L + 1)/L \rceil + 1 = p/L + 1,$$

where the last equality comes from the fact that $p$ is a multiple of $L$. But this is a contradiction. ♦

Consequently, by Claim 1, $F$ does not contain any cycle that spans $V_0$ and sets $V_i$'s, and whose length is less than or equal to $L$.

*Claim 2.* Any two cycles $C_1, C_2 \subseteq F$, $C_1 \neq C_2$, of length $L+1$ and going through $s$, cannot have a destination node in common.

*Proof.* Assume, on the contrary, that $C_1$ and $C_2$ intersect in $r$ destination nodes, $L \geq r \geq 1$. Let $q$ be the number of destination nodes covered by $C_1 \cup C_2$. First we show that $q < p$. In fact, suppose, by contradiction, that $q = p$. If $p = L$, then $|F| = L+1$. As $|C_1| = L+1$ and $|C_2 \backslash C_1| \geq 1$, we have that

$$|F| \geq |C_1| + |C_2 \backslash C_1| \geq L+2,$$

a contradiction.
So suppose that $p > L$. Then

$$q = 2L - r > L.$$

Therefore, $r < L$. But this implies that $q = p$ is not a multiple of $L$, a contradiction.
Consequently, $q < p$. Also, as $r \geq 1$, we have $q \leq 2L-1$. Moreover, observe that $|C_1 \cup C_2| \geq q+2$. This follows from the fact that $C_1 \cup C_2$ covers $q+1$ nodes and $C_1 \cup C_2$ is not a simple cycle.
Suppose, w.l.o.g., that $t_1, \ldots, t_q$ are the destination nodes covered by $C_1 \cup C_2$. Let $\tilde{\Pi} = (\tilde{V}_0, \tilde{V}_1, \ldots, \tilde{V}_{p-q})$ such that $\tilde{V}_0 = \{s, t_1, \ldots, t_q\}$, and $\tilde{V}_i = \{t_{i+q}\}$, for $i = 1, \ldots, p-q$. By the validity of the rooted-partition inequality corresponding to $\tilde{\Pi}$, we have

$$|\Delta_{\tilde{\Pi}}(G) \cap F| \geq \lceil (L+1)(p-q)/L \rceil.$$

Since $|\Delta_{\Pi}(G) \cap F| \geq |\Delta_{\tilde{\Pi}}(G) \cap F| + q + 2$, we get

$$\begin{aligned} |\Delta_{\Pi}(G) \cap F| \quad &\geq \lceil (L+1)(p-q)/L \rceil + q + 2, \\ &= p + 2 + \lceil (p-q)/L \rceil, \\ &\geq p + 2 + \lceil (p-(2L-1))/L \rceil, \\ &= p + p/L + \lceil 1/L \rceil, \\ &= p + p/L + 1. \end{aligned}$$

But this contradicts the fact that $x^F$ satisfies (4.11) with equality, and the claim is proved. ♦

*Claim 3.* The solution $F$ does not contain any chordless cycle containing $s$ of length $\geq L+2$.

*Proof.* Suppose there is a cycle $C \subseteq F$ of length $c \geq L+2$. We assume that $c$ is minimum. Suppose w.l.o.g. that $C$ goes through $s, t_1, \dots, t_{c-1}$ in this order. Since $F$ is a feasible solution for the THNDP and, by Claim 1, $F$ does not contain cycles of length less than or equal to $L$, there must exist two paths $P_1$ and $P_2$ of length $L$ joining $s$ to $t_1$ and $t_{c-1}$, respectively. Let $C_1 = \{st_1\} \cup P_1$ and $C_2 = \{st_{c-1}\} \cup P_2$.
If $c \geq L+3$, then there must also exist a chordless path $P$ from $s$ to $t_2$ of length either $L-1$ or $L$. If $P$ is of length $L-1$, then the cycles $C_1$ and $P \cup \{st_1, t_1t_2\}$ are both of length $L+1$. Since these cycles intersect in $t_1$, this contradicts Claim 2. If $P$ is of length $L$, then the cycle $P \cup \{st_1, t_1t_2\}$ is chordless and of length $L+2$. But this contradicts the minimality of $C$.
Consequently, we have that $c = L+2$. Now we are going to show that $P_1$ ($P_2$) cannot go through two nodes $t_i, t_j$, $i, j \in \{1, \dots, L+1\}$, $i < j$, such that the subpaths of $P_1$ ($P_2$) and $C$ between $t_i$ and $t_j$ are edge-disjoint. In fact, suppose for instance that the subpaths $\tilde{P}_1$ of $P_1$ and $\tilde{C}$ of $C$ between $t_i$ and $t_j$ are edge-disjoint. Let $\tilde{p}_1$ and $\tilde{c}$ be the lengths of $\tilde{P}_1$ and $\tilde{C}$, respectively. We claim that $\tilde{p}_1 = \tilde{c}$. Indeed, if $\tilde{p}_1 > \tilde{c}$, then, by replacing $\tilde{P}_1$ by $\tilde{C}$ in $P_1$, we get a path of length $\leq L-1$ between $s$ and $t_1$, a contradiction. On the other hand, if $\tilde{c} > \tilde{p}_1$, then, by replacing $\tilde{C}$ by $\tilde{P}_1$ in $C$, we obtain a cycle, say $C'$, of length $\leq L+1$. Since, by Claim 1, $F$ does not contain a cycle of length $\leq L$, it follows that $C'$ is of length exactly $L+1$. Since $C_1$ and $C'$ have $t_1$ in common, this contradicts Claim 2.
Therefore $\tilde{p}_1 = \tilde{c}$. But this is still a contradiction since $C_1$ and the cycle obtained from $C_1$ by replacing $\tilde{P}_1$ by $\tilde{C}$ are both of length $L+1$ and have destination nodes in common.
In consequence, if $t_{l_1}$ (resp. $t_{l_2}$) is the first node of $C$ met by $P_1$ (resp. $P_2$), then $P_1$ (resp. $P_2$) contains the subpath of $C$ between $t_{l_1}$ and $t_1$ (resp. $t_{l_2}$ and $t_{L+1}$). See Fig. 4.6 for an illustration. Moreover, we have that $l_1 < l_2$, for otherwise, the cycles $C_1$ and $C_2$ would intersect in some destination nodes, which by Claim 2 is impossible.

Let $r = l_2 - l_1 - 1$. Observe that $r$ is the number of internal nodes of the subpath of $C$ between $t_{l_1}$ and $t_{l_2}$. Also note that $r < L$. Now let $\overline{\Pi} = (\overline{V}_0, \overline{V}_1, \dots, \overline{V}_{p-2L-r})$ be the partition of $N$ obtained from $\Pi$ by gathering the nodes of $C_1 \cup C_2 \cup C$ into $\overline{V}_0$. Note that $|E(\overline{V}_0)| \geq 2(L+1)+r+1$. By the validity of the rooted-partition induced

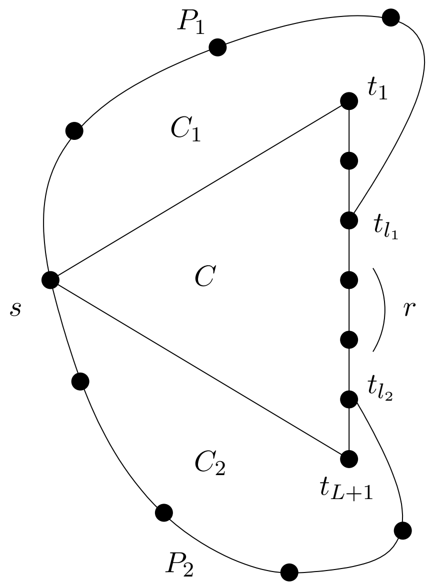


Figure 4.6: An example of an $(L+2)$-cycle $C$ and the two associated $(L+1)$-cycles $C_1$ and $C_2$

by $\overline{\Pi}$, we have

$$\begin{aligned}|\Delta_{\overline{\Pi}}(G) \cap F| &\geq p - 2L - r + \lceil (p - 2L - r)/L \rceil, \\ &= p - 2L - r + p/L - 2.\end{aligned}$$

Remark that this remains true even if $p - 2L - r = 0$, that is if $\overline{V}_0 = N$, since in this case $\Delta_{\overline{\Pi}}(G) \cap F$ is empty and the righthand side of (4.11) is equal to 0. So,

$$|\Delta_{\Pi}(G) \cap F| \geq |\Delta_{\overline{\Pi}}(G) \cap F| + 2(L+1) + r + 1 \geq p + p/L + 1,$$

a contradiction, which finishes the proof of the claim. ♦

From Claims 1 and 3, it follows that the only cycles induced by $F$ are of length exactly $L+1$. By Claim 2, these cycles do not contain destination nodes in common. Therefore $F$ consists of $p/L$ cycles of length $L+1$ and having only $s$ in common. Since each of these cycles uses exactly two edges of $[s, T]$, this yields that $|F \cap [s, T]| = 2p/L$, and the proof of the theorem is complete. □

Given a graph $G = (N, E)$, and a node subset $V \subseteq N$ such that $s \in V$, we say that $V$ satisfies Property $(\pi)$ if one of the following holds :

i) $|V| \geq 4$,

ii) $|V| = 2, 3$ and $V$ contains no destination node of $s$ (two Steiner nodes),

iii) $|V| = 3$, $V$ contains two destination nodes $t, t'$ of $s$ (no Steiner node), and at least two of the sets $[s, t], [s, t'], [t, t']$ contain parallel edges,

iv) $|V| = 3$, $V$ contains one destination node $t$ of $s$ (and hence one Steiner node $u$), and either $|[s, t]| \geq 2$, or $|[s, u]| \geq 2$ and $|[u, t]| \geq 2$,

v) $|V| = 2$, $V$ contains one destination node $t$ of $s$ (no Steiner node), and $|[s, t]| \geq 2$,

vi) $|V| = 1$.

**Theorem 4.5.4.** *Let $L = 2$, and $G = (N, E)$ be a complete graph with one source node $s$ and at least $p \geq 2$ destination nodes associated to $s$. Let $\Pi = (V_0, V_1, \dots, V_p)$ be a partition of $N$ such that $s \in V_0$ and $t_i \in V_i$ for $i = 1, \dots, p$.*
*Then, the rooted-partition inequality (4.11) defines a facet if and only if*

*(i) $p$ is odd,*

*(ii) $|V_i| = 1$ for all $i = 1, \dots, p$,*

*(iii) $V_0$ satisfies Property ($\pi$).*

*Proof. Necessity.*
(i) This follows from Theorem 4.5.3.

(ii) Suppose that $V_i$ contains a further node $u$. Then consider the partition $(V'_0, \dots, V'_{p+1})$ such that $V'_l = V_l$, for $l \in \{0, 1, \dots, p\} \backslash \{i\}$, $V'_i = V_i \backslash \{u\}$, and $V'_{p+1} = \{u\}$. We have that the rooted-opt-partition inequality (valid for $L = 2$) induced by this partition and some edge $e \in [V'_j, V'_{p+1}]$, $j \in \{0, 1, \dots, p\} \backslash \{i\}$, dominates inequality (4.11). Therefore, the latter one cannot be facet-defining.

(iii) Let us suppose that $V_0$ does not satisfy Property ($\pi$) and that the rooted-partition inequality is, nevertheless, facet-defining. We have either $|V_0| = 2$ or $|V_0| = 3$. Suppose first that $|V_0| = 2$ and let $t$ be the other node in $V_0$ along with $s$. By hypothesis, we also have that $t$ is a destination node and that $|[s, t]| = 1$. Consider a solution $F$ such that $x^F$ satisfies the rooted-partition inequality with equality. Thus $F$ contains exactly $\lceil 3p/2 \rceil$ edges in $\Delta_\Pi(G)$, which, by the validity of the inequality, are all necessary to

link the demand nodes other than $t$ to $s$. We will show that $F$ necessarily contains $st$, which will yield a contradiction to the fact that the rooted-partition induces a facet. Suppose, on the contrary, that $F$ does not contain the edge $st$. Then there must be at least two edges in $\Delta_\Pi(G)$ linking $t$ to the other nodes. However, these edges are not useful to the 2-paths of the demand nodes other than $t$. Consequently, we have at least $\lceil 3p/2 \rceil + 2$ edges from $\Delta_\Pi(G)$ in $F$, a contradiction.
Now suppose that $|V_0| = 3$, and that $V_0$ contains two destination nodes, $t$ and $t'$. By hypothesis, we have that at most one of the sets $[s,t]$, $[s,t']$, $[t,t']$ contains parallel edges. Suppose first that all these sets contain a single edge. Then, by putting $t$ and $t'$ in two additional partition subsets, we get a new rooted-partition inequality, which dominates the previous one, a contradiction. If there are exactly one set of parallel edges and two single edges, we can suppose w.l.o.g. that $|[s,t]| = 1$. We can show, in a similar way as when $|V_0| = 2$, that $st$ belongs to any solution whose incidence vector satisfies (4.11) with equality, which yields again a contradiction.
Finally, if $|V_0| = 3$, and $V_0$ contains one destination node $t$ and one Steiner node $u$, we have that $|[s,t]| = 1$, and $|[s,u]| = 1$ or $|[u,t]| = 1$. Once again, one can show that $st$ must belong to any solution in the face.

*Sufficiency.* Suppose that (i), (ii) and (iii) hold. As $p \geq 2$ and $p$ is odd, we have that $p \geq 3$. Let us denote inequality $x(\Delta_\Pi(G)) \geq \lceil 3p/2 \rceil$ by $ax \geq \alpha$ and let $bx \geq \beta$ be a facet-defining inequality of $P_G(D,2)$ such that

$$\{x^F \in P_G(D,2) \mid ax^F = \alpha\} \subseteq \{x^F \in P_G(D,2) \mid bx^F = \beta\}.$$

We will show that $b = \rho a$ for some $\rho > 0$.

Consider the edge sets
$F_0 = \{st_i, i = 1,\dots,p\} \cup \{t_i t_{i+1}, i = 1,3,\dots,p-2\} \cup \{t_{p-1}t_p\} \cup E(V_0)$,
$F_e = (F_0 \backslash \{t_{p-1}t_p\}) \cup \{e\}$,
for some $e \in \delta(t_p) \backslash \{st_p\}$.

It is not hard to see that $F_0$ and $F_e$ belong to $S_a(G)$. Thus $bx^{F_0} = bx^{F_e} = \beta$, which implies that

$$b(e) = b(t_{p-1}t_p), \text{ for all } e \in \delta(t_p) \backslash \{st_p\}. \quad (4.16)$$

Furthermore, let $F_0' = (F_0 \backslash \{st_p\}) \cup \{t_1 t_p\}$. Since $ax^{F_0'} = \alpha$, and hence $bx^{F_0'} = \beta$, we obtain that $b(st_p) = b(t_1t_p)$. This together with (4.16) yields

$$b(e) = b(e'), \text{ for all } e, e' \in \delta(t_p).$$

By exchanging the roles of $t_p$ and the $t_i$'s, $i \neq p$, we get

$$b(e) = \rho, \text{ for all } e \in \Delta_\Pi(G), \tag{4.17}$$

for some $\rho \in \mathbb{R}$.

Now suppose that $E(V_0) \neq \emptyset$ (hence case vi) of Property ($\pi$) does not hold), and let $f \in E(V_0)$. If one case among i)-iv) of ($\pi$) holds, then $F_0'' = F_0 \backslash \{f\}$ is still in $S_a(G)$. Thus its incidence vector satisfies $bx \geq \beta$ as equality, and we get
$0 = bx^{F_0} - bx^{F_0''} = b(f)$.

Now assume that case v) of ($\pi$) holds. Let $t$ be the terminal node of $V_0$ different from $s$. By Property ($\pi$) v), $|[s,t]| \geq 2$. Let $F_1 = (F_0 \backslash \{t_{p-1}t_p, f\}) \cup \{tt_p\}$. Clearly, $F_1 \in S(G)$. As $ax^{F_1} = ax^{F_0} = \alpha$, it follows that
$0 = bx^{F_0} - bx^{F_1} = b(f) + b(t_{p-1}t_p) - b(tt_p)$.
Since by (4.17) $b(t_{p-1}t_p) = b(tt_p)$, we have that

$$b(f) = 0.$$

Consequently, we obtain

$$\begin{aligned} b(e) &= 0, \text{ for all } e \in E(V_0), \\ b(e) &= \rho, \text{ for all } e \in \Delta_\Pi(G). \end{aligned}$$

Since $bx \geq \beta$ is different from a trivial inequality, we have by Lemma 4.5.2 that $\beta > 0$, and hence $\rho > 0$. Therefore, $b = \rho a$ with $\rho > 0$. □

We now present necessary and sufficient conditions for the double cut inequalities to be facet-defining when $L = 2$. We consider first the case where $s \in V_0$.

**Theorem 4.5.5.** *Let $L = 2$ and $G = (N, E)$ be a complete graph with $|N| \geq 4$, one source node $s$ and $d$ destination nodes $T = \{t_1, \ldots, t_d\}$. Let $(V_0, V_1, V_2, V_3)$ be a partition of $N$ such that $s \in V_0$, $t_2 \in V_1$ and $t_1 \in V_3$. Let $e \in [V_1, V_2]$. Then the double cut inequality (4.6) is facet-inducing for $P_G(D, 2)$ if and only if* *(i)* $|V_0| = 1$,

*(ii)* $|V_1| = 1$,

*(iii)* $d = 2$,

*(iv)* *if* $v_2$ *is the end node of* $e$ *in* $V_2$ *and* $|[t_2, v_2]| \geq 2$, *then* $|[t_1, v_2]| \geq 2$ *and* $|[s, v_2]| \geq 2$.

*Proof.* Let us denote inequality (4.6) by $ax \geq \alpha$ and let $S_a(G)$ be the induced face.

*Necessity.*

(i) Suppose, on the contrary, that inequality (4.6) is facet-defining while $|V_0| \geq 2$. Let $v_0 \in V_0 \backslash \{s\}$. We are going to show that any feasible solution $F \in S_a(G)$ does not intersect $[v_0, V_2]$, which contradicts Lemma 4.5.1. Suppose that this is not the case and that there exists a feasible solution $F$ containing $f \in [v_0, V_2]$ and such that $x^F$ satisfies (4.6) as equation. First, note that, as $f$ belongs to the double cut, $F$ contains exactly two more edges of it. On the other hand, since $f$ is incident neither to $s$, nor to $t_1$ (resp. $t_2$), $f$ is not useful to the two 2-paths between $s$ and $t_1$ (resp. $t_2$) that $F$ must contain. However, it is not possible to build these four paths with only two more edges from the double cut.

(ii) Suppose on the contrary that inequality (4.6) is facet-defining while $|V_1| \geq 2$. Hence, since the graph $G$ is complete, we have that $|[V_1, V_2]| \geq 2$. Suppose first that $e \in [t_2, V_2]$. Then, we have that any edge $f \in [V_1 \backslash \{t_2\}, V_2]$ belongs to the double cut. However, $f$ is not useful for building 2-paths between $s$ and $t_2$, or $s$ and $t_1$. Therefore, any feasible solution $F \in S_a(G)$ cannot contain $f$. Thus, by Lemma 4.5.1, we have a contradiction.
Suppose now that $e \in [V_1 \backslash \{t_2\}, V_2]$. In that case, any edge $f \in [t_2, V_2]$ is in the double cut. Consider a solution $F \in S_a(G)$ that contains $f$. If $f$ is not used in a 2-$st_2$-path of $F$, then $f$ does not belong to any of the 2-paths between $s$ and $t_2$, and $s$ and $t_1$, and therefore we get the same contradiction as before. As a consequence, $f$ must belong to a 2-$st_2$-path of $F$ of the form $(s, v_2, t_2)$ with $v_2 \in V_2$. As this path uses two edges of the double cut, $F$ can only contain one more edge of it for the other 2-$st_2$-path and for at least one of the two 2-$st_1$-paths (given that $sv_2$ can be used together with an edge of $[v_2, t_1]$ to form a 2-$st_1$-path). But this is also impossible.

(iii) By definition of the double cut, we already have that $d \geq 2$. Let us show now that $d \leq 2$ when the corresponding inequality is facet-defining. Suppose, on the contrary, that the double cut inequality is facet-defining while there exists a third destination of $s$, say $t_3$. By (i) and (ii), we have that $t_3 \in V_2 \cup V_3$. Consider a solution $F \in S_a(G)$ not containing $e$. Therefore, the two 2-$st_2$-paths of $F$ are only constituted of edges in the double cut. First, it is clear that these paths cannot be both of length 2. If one of the paths is of length 1 and the other is of length 2, we already have three edges taken by $F$ in the double cut, while we still need to construct at least one 2-path for one of the two destination nodes $t_1, t_3$. Clearly, this is impossible. Finally, if both 2-$st_2$-paths are of length 1, we can still take one more edge in the double cut. But, this time, we need to link both $t_1$ and $t_3$ to $s$ by two 2-paths. Once again, this is not possible. We get therefore that any solution $F \in S_a(G)$ does contain $e$, which contradicts Lemma 4.5.1.

(iv) Suppose that the double cut inequality is facet-defining and that $|[t_2, v_2]| \geq 2$. Therefore, any edge $e' \in [t_2, v_2] \backslash \{e\}$ belongs to the double cut. By Lemma 4.5.1, there must exist a feasible solution $F$ in $S_a(G)$ containing $e'$. Moreover, $e'$ must belong to a 2-$st_2$-path of $F$. This path is then of the form $(sv_2, e')$. As $sv_2$ and $e'$ belong to the double cut, only one more edge of it must be used to form a second 2-$st_2$-path and one 2-$st_1$-path (the other one using the edge $sv_2$ plus an edge in $[v_2, t_1]$). Clearly, the only possibility to do that is to go once again through $v_2$. As a consequence, we obtain that both $[s, v_2]$ and $[v_2, t_1]$ contain parallel edges.

*Sufficiency.*
Suppose that (i), (ii), (iii) and (iv) hold. Let $v_2$ be the end node of $e$ in $V_2$. Let $bx \geq \beta$ be a facet-defining inequality of $P_G(D, 2)$ such that

$$\{x^F \in P_G(D,2) \,|\, ax^F = \alpha\} \subseteq \{x^F \in P_G(D,2) \,|\, bx^F = \beta\}.$$

As before, we will show that $b = \rho a$ for some $\rho > 0$.

Consider the solution $F = \{st_1, st_2, t_1t_2\}$. Clearly, its incidence vector $x^F$ satisfies $ax = \alpha$. Note that we can add to $F$ all the edges in $E(V_2 \cup V_3) \cup \{e\}$ while $x^F$ still verifies $ax = \alpha$. Consequently, we get

$$b(f) = 0, \text{ for all } f \in E(V_2 \cup V_3) \cup \{e\}. \tag{4.18}$$

Now, let $F^* = (F \backslash \{t_1t_2\}) \cup \{sv_2, e, v_2t_1\}$. Clearly, $F^* \in S_a(G)$. Hence $bx^{F^*} = \beta$. As by (4.18) $b(e) = b(v_2t_1) = 0$, this yields

$$b(t_1t_2) = b(sv_2). \tag{4.19}$$

The same holds if we replace in $F^*$ $st_1$ (or $st_2$) by $t_1t_2$. Therefore

$$b(st_1) = b(t_1t_2) = b(st_2). \tag{4.20}$$

From (4.19) and (4.20), we get

$$b(st_1) = b(st_2) = b(t_1t_2) = b(sv_2).$$

As these edges are arbitrary edges from $[s, t_1]$, $[s, t_2]$, $[t_1, t_2]$, $[sv_2]$, respectively, we obtain that

$$b(f) = \rho, \text{ for all } e \in [s, t_1] \cup [s, t_2] \cup [t_1, t_2] \cup [s, v_2], \tag{4.21}$$

for some scalar $\rho$.

Now consider the solution
$F_v = \{sv_2, e, v_2t_1, sv, vt_2, vt_1\}$
for $v \in (V_2 \cup V_3)\backslash\{v_2, t_1\}$. It is not hard to see that $F \in S_a(G)$. Therefore $bx^{F_v} = \beta$. As $bx^{F^*} = \beta$, using (4.18) we obtain that

$$b(st_1) + b(st_2) = b(sv) + b(vt_2). \tag{4.22}$$

Also consider
$F'_v = (F_v\backslash\{vt_2\}) \cup \{st_2\}$.
Since $F'_v \in S_a(G)$, and hence $bx^{F'_v} = \beta$, we get

$$b(st_2) = b(vt_2). \tag{4.23}$$

From (4.22) and (4.23), it follows that

$$b(st_1) = b(sv). \tag{4.24}$$

From (4.23) and (4.24) together with the fact that $sv$ (resp. $vt_2$) is an arbitrary edge between $s$ and $v$ (resp. $v$ and $t_2$), by (4.21) it follows that

$$b(f) = \rho, \text{ for all } f \in [\{s, t_2\}, V_2 \cup V_3]\backslash[v_2, t_2]. \tag{4.25}$$

If there are edges in $[v_2, t_2]\backslash\{e\}$, we still need to prove that their coefficient in $b$ is equal to $\rho$. Suppose this is the case and let $v_2t_2$ be an edge of $[v_2, t_2]\backslash\{e\}$. Then, by (iv), the edge set $\tilde{F} = \{sv_2, g, e, v_2t_2, v_2t_1, h\}$ exists. Here, $g$ and $h$ are edges parallel to $sv_2$ and $v_2t_1$, respectively. Clearly, its incidence vector $x^{\tilde{F}}$ satisfies $ax = \alpha$. Thus $bx^{\tilde{F}} = \beta$. As $bx^{F^*} = \beta$, we obtain

$$b(g) + b(v_2t_2) = 2b(st_2). \tag{4.26}$$

From (4.21) and the fact that $v_2t_2$ is an arbitrary edge of $[v_2, t_2]\backslash\{e\}$, we get

$$b(f) = \rho, \text{ for all } f \in [v_2, t_2]\backslash\{e\}. \tag{4.27}$$

From (4.18),(4.21),(4.25),(4.27), we have $b = \rho a$. As $bx \geq \beta$ is a facet-defining inequality different from a trivial one, by Lemma 4.5.2, it follows that $\rho > 0$. □

The following theorem gives a similar result for the double cut inequalities based on partitions with $s \in V_3$. Its proof is along the same line as that of Theorem 4.5.5.

**Theorem 4.5.6.** *Let $L = 2$ and $G = (N, E)$ be a complete graph with $|N| \geq 6$, one source node $s$ and $d$ destination nodes $T = \{t_1, \dots, t_d\}$. Let $(V_0, V_1, V_2, V_3)$ be a partition of $N$ such that $t_1 \in V_0$, $t_2 \in V_1$ and $s \in V_3$. Let $e \in [V_1, V_2]$. Then the double cut inequality (4.6) is facet-inducing for $P_G(D, 2)$ if and only if*

*(i)* $|V_0| = 1$,

*(ii)* $|V_1| = 1$,

*(iii) if $v_2$ is the end node of $e$ in $V_2$ and $|[t_2, v_2]| \geq 2$, then $|[s, v_2]| \geq 2$.*

For $L = 3$, we have also investigated necessary conditions for the double cut inequalities (4.7) to be facet-defining. More precisely, we have the following.

**Theorem 4.5.7.** *When $L = 3$, if inequality (4.7) defines a facet of $P_G(D, 3)$, then*

*(i)* $|V_0| = 1$,

*(ii)* $|V_2| = 1$,

*(iii)* $|V_4| \leq 2$.

*Proof.* (i) Let $s \in V_0$ and suppose that $|V_0| \geq 2$. Let $v_0 \in V_0 \backslash \{s\}$. Consider a feasible solution satisfying the double cut inequality with equality and containing an edge $v_0v_2$ with $v_2 \in V_2$. As this edge has a coefficient 1 in the double cut inequality, it must be used in the 3-paths linking $t_1, t_2$ to $s$. Suppose first that it belongs to a path of the form $s, v_2, v_0, t_1$ (resp. $t_2$). Clearly, since three edges of the double cut are already used by this path, it is impossible to construct three other 3-paths with only these edges plus those of coefficient 0, hence a contradiction. Now, suppose that we have rather a path of the form $s, v_0, v_2, t_1$ (resp. $t_2$). Without loss of generality, we can suppose that the third edge of this path is $e$. Hence, we may only use two more edges in the double cut to construct another 3-path to $t_1$ (resp. $t_2$) and two 3-paths to $t_2$ (resp. $t_1$). Clearly, this is impossible, which contradicts Lemma 4.5.1. As a consequence, we have that $|V_0| = 1$. A similar proof can be done when $s \in V_4$.

(ii) Let $s \in V_0$ and suppose that $|V_2| \geq 2$. Let $v_2, v_2' \in V_2$ such that $e \in \delta(v_2)$. Consequently, all the edges in $[v_2', V_1 \cup V_4]$ have a coefficient 1 in the double cut inequality. Now, there must exist a feasible solution satisfying this inequality as an equality and containing the edge $v_2't_1$. Clearly, this edge must belong to a 3-$st_1$-path. Moreover, this path must contain an edge in $\delta(V_0)$, which is of coefficient 1. We consider two cases. If the path uses an edge in $[V_0, V_1 \cup V_4]$, it must then contain an edge of $[V_1 \cup V_4, v_2']$ and no more edge in the double cut can be used to construct a second 3-$st_1$-path, which is impossible. If the path uses an edge $f \in [V_0, V_2 \cup V_3]$, its third edge (if any) is of coefficient 0. Moreover, in the best case, that is if $e$ is incident to $t_2$, it will be possible to complete $f$ by edges of coefficient 0 to form a 3-$st_2$-path. Now, we have to construct a second 3-$st_2$-path using only one edge of the double cut. Clearly, the only possibility is to take an edge in $[V_0, V_1]$. But then, it is impossible to complete it by edges of coefficient 0 to make a second 3-$st_1$-path. A similar proof can be done when $s \in V_4$.

(iii) Let $s \in V_0$. We are going to show that, if there exists $w \in V_4 \backslash \{t_1\}$, then $e \in [w, v_2]$ (recall that $V_2 = \{v_2\}$). As a consequence, by unicity of $e$, we will have that $|V_4| \leq 2$. By contradiction, let $w \in V_4 \backslash \{t_1\}$ and suppose that $e$ is not incident to $w$. Then, the edge $wv_2$ is of coefficient 1 and there must exist a solution containing it plus two other edges in the double cut. As all the edges in $\delta(V_0)$ have a coefficient 1, we are obliged to take the two additional edges in this subset. Moreover, since at most one edge of $\delta(V_1)$ (i.e. $e$) is of coefficient 0, at least one of these two edges belongs to $[V_0, V_1]$. On the other hand, as the edge $wv_2$ must be used by a 3-path, the other of these two edges must be in $[s, v_2]$ or $[s, w]$. But then, in both cases, it is clearly impossible to complete the edge in $[V_0, V_1]$ by edges of coefficient 0 to form a second 3-$st_1$-path. A similar proof can be done when $s \in V_4$. □

We have also investigated the conditions under which the triple path-cut inequalities are facet-defining when $L = 2$. Note that, in the case of rooted demands, we simply pose $s_1 = s_2$.

**Theorem 4.5.8.** *Let $L = 2$ and $G = (N, E)$ be a complete graph with $|N| \geq 5$, one source node $s$ and $d$ destination nodes $T = \{t_1, \dots, t_d\}$. Let $(V_0, V_1, V_2, V_3, V_4)$ be a partition of $N$ such that $s \in V_0$, $t_1 \in V_3$ and $t_2 \in V_4$. Let $e \in [V_1, V_3]$. Then the triple path-cut inequality (4.9) is facet-defining for $P_G(D, 2)$ if and only if*

*(i)* $|V_0| = 1$,

*(ii)* $|V_3| = 1$,

*(iii)* $|V_4| = 1$,

*(iv) if $v_1$ is the end node of $e$ in $V_1$ and $|[v_1, t_1]| \geq 2$, then $|[s, v_1]| \geq 2$.*

*(v) if $|V_1| \leq 2$ and $V_1 \subset T$, then there exists $t \in V_1 \cap T$ such that $|[s, t]| \geq 2$.*

*(vi) if $|V_1| = 2$ and $V_1 \cap T = \{t_3\}$, then at least one of the following holds :*

- *$e$ is incident to $t_3$,*
- $|[s, V_1 \backslash \{t_3\}]| \geq 2$,
- $|[s, t_3]| \geq 2$.

*Proof.* As before, let us denote inequality (4.9) by $ax \geq \alpha$. We will consider $a(e)$ as a weight on $e$. Hence, any solution $F$ of $S_a(G)$ will have a weight $a(F) = 3$.

*Necessity.* Assume that $ax \geq \alpha$ defines a facet of $P_G(D, 2)$.

(i) Suppose, by contradiction, that $|V_0| \geq 2$. Let $v_0 \in V_0 \backslash \{s\}$. Consider a solution $F \in S_a(G)$ containing an edge $f \in [v_0, V_2]$. Since $a(f) = 2$, $F$ can only contain one more edge with weight 1 in the triple path-cut, say $f'$. However, as $f$ is incident neither to $s$, nor to $t_1$ (resp. $t_2$), $f$ cannot be used by any of the two 2-$st_1$- (resp. 2-$st_2$-) paths of $F$. Clearly, these four paths cannot be constructed only with $f'$ along with the edges of weight zero in the triple path-cut. Consequently, we have that any solution $F$ of $S_a(G)$ does not contain $f$, contradicting Lemma 4.5.1.

(ii) Suppose on the contrary that $|V_3| \geq 2$. Let $v_3 \in V_3 \backslash \{t_1\}$. Suppose first that $e$ is incident to $v_3$. Then, any edge of $[V_1, t_1]$ is in the triple path-cut. Consider a solution $F \in S_a(G)$ containing an edge $f = v_1 t_1 \in [V_1, t_1]$. Note that we can complete $f$ by an edge $sv_1$ (of weight zero) to form a first 2-$st_1$-path. However, with only an additional weight of 2 in the triple path-cut, it is impossible to form another 2-$st_1$-path and two 2-$st_2$-paths, hence a contradiction.
Suppose now that $e$ is not incident to $v_3$. Therefore any edge $f' \in [V_1, v_3]$ is of weight 1. We claim that there does not exist a feasible solution $F'$ of weight 3 containing such an edge $f'$. Indeed, as $f'$ is incident neither to $s$, nor to $t_1$ (resp. $t_2$), $f'$ is not useful to the two 2-$st_1$- (resp. 2-$st_2$-) paths of $F'$. So, we can only use either two more edges of weight 1, or one more edge of weight 2, in order to build these four paths. Obviously, this is impossible. Therefore, $f' \notin F'$. But this contradicts Lemma 4.5.1.

(iii) The proof is similar to the second part of (ii).

(iv) By (ii), $|V_3| = \{t_1\}$ and hence $e \in [v_1, t_1]$. Suppose that $|[v_1, t_1]| \geq 2$. Let $\bar{e}$ be an edge parallel to $e$, and let $F$ be a solution of $S_a(G)$ containing $\bar{e}$. By definition, $\bar{e}$ has a weight 1. Observe that $\bar{e}$ must be used in one 2-$st_1$-path, of the form $(s, v_1, t_1)$. Otherwise $\bar{e}$ would not be used by any 2-$st_1$- and 2-$st_2$-path in $F$, and thus $F$ could not have a weight 3, which is a contradiction. Moreover, $e$ must belong to the second 2-$st_1$-path of $F$. If not, one could indeed replace $\bar{e}$ by $e$ and get a solution of weight 2, a contradiction. This path must also be of the form $(s, v_1, t_1)$ and therefore $|[s, v_1]| \geq 2$.

(v) Suppose that $|V_1| = 2$ and that $V_1 = \{t_3, t_4\}$ (if $|V_1| = 1$, the proof is similar). W.l.o.g., let $t_3$ be the end node of $e$ in $V_1$. Assume, by contradiction, that $[s, t_3] = \{st_3\}$ and that $[s, t_4] = \{st_4\}$. Let $F \in S_a(G)$ not containing $st_3$. Then the two 2-$st_3$-paths of $F$ must be of length exactly 2. Let $(s, u, t_3)$ and $(s, v, t_3)$ be these two paths. Clearly, the edges $ut_3$ and $vt_3$ are not useful for linking $t_1$ or $t_2$ to $s$. Therefore, $ut_3$ and $vt_3$ must be both of weight zero and thus we get that $u, v \in \{t_1, t_4\} \cup V_2$. Remark that we can consider $u = t_4$. In fact, any 2-$st_3$-path going through $t_4$ is of weight zero.
Now, if $v \in V_2$, then the 2-$st_3$-path $(sv, vt_3)$ is of weight 2. Also we can use edge $sv$ in a 2-$st_1$-path and in a 2-$st_2$-path along with edges of weight zero. But in this case, it remains to construct one more 2-path between $s$ and $t_1$, and one more 2-path between $s$ and $t_2$. However, this is not possible using only one more edge of weight 1. (Note that edge $e$ cannot be used here.)
If $v = t_1$, then the 2-$st_3$-path $(sv, vt_3)$ is this time of weight 1. As $t_1$ is already covered by $sv = st_1$, there must exist one more 2-$st_1$-path and two 2-$st_2$-paths in $F$ with a total

weight $\leq 2$. Clearly, the only possibility for that is to use edges $st_2$ and $t_1t_2$. But there must also exist a 2-path in $F$ between $s$ and $t_4$ (besides $st_4$). As $a(st_2) + a(t_1t_2) = 2$, this path must only use edges of weight zero. As $|[s, t_4]| = 1$, this is impossible.
So, in both cases, we obtain that $st_3 \in F$, contradicting Lemma 4.5.1.

(vi) The proof is similar to that of (v).

*Sufficiency.*
Suppose that (i), (ii), (iii), (iv), (v) and (vi) hold. Let $v_1$ be the end node of $e$ in $V_1$. Let $bx \geq \beta$ be a facet defining inequality of $P_G(D, 2)$ such that

$$\{x^F \in P_G(D, 2) \,|\, ax^F = \alpha\} \subseteq \{x^F \in P_G(D, 2) \,|\, bx^F = \beta\}.$$

As before, we will show that $b = \rho a$ for some $\rho > 0$.

By (i), (ii) and (iii), we have that $V_0 = \{s\}$, $V_3 = \{t_1\}$, $V_4 = \{t_2\}$, and hence $T \backslash \{t_1, t_2\} \subseteq V_1 \cup V_2$. Consider the solution
$F = \{st_1, st_2, t_1t_2\} \cup \overline{E}$,
where,
$\overline{E} = \{f \in E \,|\, a(f) = 0\}$.
Clearly, $F \in S_a(G)$. First we show that

$$b(f) = 0, \text{ for all } f \in \overline{E}. \tag{4.28}$$

If $T \cap (V_1 \cup V_2) = \emptyset$, then $F \backslash \overline{E}$ is a solution of the THNDP. Hence $F \backslash \{f\} \in S(G)$ for all $f \in \overline{E}$, and (4.28) holds.

Now suppose that $T \cap (V_1 \cup V_2) \neq \emptyset$. Assume first that $T \cap V_1 = \emptyset$, and therefore $T \cap V_2 \neq \emptyset$. Let $t$ be a destination node of $V_2$. Observe that, in $F$, $t$ is connected to $s$ by three 2-paths, namely $(sv, vt)$, $(st_1, t_1t)$, $(st_2, t_2t)$, where $v \in V_1$. Thus, by deleting from $F$ any edge among $\{sv, vt, t_1t, t_2t\}$, the resulting solution is still feasible. As $v$ and $t$ are arbitrary in $V_1$ and $V_2$, this implies that $F \backslash \{f\} \in S(G)$ for all $f \in [s, V_1] \cup [V_1 \cup \{t_1, t_2\}, V_2 \cap T]$. But we can also easily remark that, in this case, the solution obtained from $F$ by deleting any edge of $\overline{E} \backslash ([s, V_1] \cup [V_1 \cup \{t_1, t_2\}, V_2 \cap T])$ is still feasible. Therefore, we have that (4.28) holds.

Now suppose that $T \cap V_1 \neq \emptyset$. Consider first the case where $T \cap V_2 = \emptyset$. We consider two cases.

**Case 1:** $|V_1| \geq 3$. Then it is clear that $F \backslash \{f\} \in S(G)$ for all $f \in E(\{s\} \cup V_1)$. Moreover, as $T \cap V_2 = \emptyset$, we also have that $F \backslash \{f\} \in S(G)$ for all $f \in \overline{E} \backslash E(\{s\} \cup V_1)$.
**Case 2:** $|V_1| \leq 2$. Suppose that $|V_1| = 2$ (the case where $|V_1| = 1$ is similar). Thus $V_1$ contains either one or two destination nodes.
First suppose that $V_1 = \{t_3, t_4\} \subset T$ and that $t_3$ is the end node of $e$ in $V_1$. By (v), we have that at least one of the sets $[s, t_3]$ and $[s, t_4]$ contains two edges.
If $|[s, t_4]| \geq 2$, then
$F' = \{st_1, st_2, t_1t_2, e, st_3, t_3t_4\} \cup [s, t_4]$
is a solution of $S(G)$. Therefore $F \backslash \{f\} \in S(G)$ for all $f \in \overline{E} \backslash (\{e, st_3, t_3t_4\} \cup [s, t_4])$. Moreover, it is easy to see that $F' \backslash \{f'\} \in S(G)$, for all $f' \in \{e, st_3, t_3t_4\} \cup [s, t_4]$. Hence, equation (4.28) holds.
If $|[s, t_4]| = 1$, then $|[s, t_3]| \geq 2$. Consider the solutions
$F_1 = \{sv_2, t_2t_3\} \cup \overline{E}$,
$F_2 = \{sv_2, t_2t_3\} \cup \tilde{E}$,
where $v_2 \in V_2$ and $\tilde{E} = [s, t_3] \cup \{st_4, t_3t_4, t_3v_2, t_4v_2, e, v_2t_1, v_2t_2\}$. We have that $F_1, F_2 \in S_a(G)$. Hence $b(f) = 0$ for all $f \in \overline{E} \backslash \tilde{E}$. Also, it is not hard to see that $F_2 \backslash \{f\} \in S(G)$ for all $f \in \tilde{E} \backslash \{e, v_2t_1, v_2t_2\}$. Finally, we have that $F \backslash \{f\} \in S(G)$, where $F$ is the solution introduced above and $f \in \{e, v_2t_1, v_2t_2\}$. Overall, we obtain

$$b(f) = 0, \text{ for all } f \in \overline{E},$$

and hence (4.28) holds.
If $|V_1 \cap T| = 1$, equation (4.28) can be shown in a similar way, using this time condition (vi).

Finally, the case where $T \cap V_2 \neq \emptyset$ can be treated using similar ideas as before, and we obtain that (4.28) also holds in this case.

Now we show that $b(f) = \rho$ for all $f \in E \backslash (\overline{E} \cup [s, V_2])$, and $b(f) = 2\rho$ for all $f \in [s, V_2]$, for some $\rho \in \mathbb{R}$. Consider the solution $F$ introduced above, which belongs to $S_a(G)$. Let $F^* = (F \backslash \{t_1t_2\}) \cup \{v_1t_2\}$. Clearly, $F^* \in S_a(G)$, and hence $bx^{F^*} = bx^F = \beta$. This yields

$$b(v_1t_2) = b(t_1t_2). \tag{4.29}$$

The same holds if we replace in $F^*$ the edges $st_1, st_2$ by $sv_2$, with $v_2 \in V_2$, and thus we get

$$b(sv_2) = b(st_1) + b(st_2). \tag{4.30}$$

Consider now the edge sets
$F_1 = \{st_1, v_1t_2, t_1t_2\} \cup \overline{E}$,
$F_2 = \{st_2, v_1t_2, t_1t_2\} \cup \overline{E}$.
We have that $F_1, F_2 \in S_a(G)$ and hence $bx^{F_1} = bx^{F_2} = \beta$. As $bx^F = \beta$, we have

$$b(st_2) = b(v_1t_2) = b(st_1). \tag{4.31}$$

This together with (4.30) yield

$$b(sv_2) = 2b(st_1). \tag{4.32}$$

As $v_2$ is an arbitrary node in $V_2$, it follows that

$$b(f) = 2b(st_1), \text{ for all } f \in [s, V_2]. \tag{4.33}$$

Consider now the edge sets $\{v_1t_2, v_1't_1, st_2\} \cup \overline{E}$ and $\{st_1, v_1't_2, st_2\} \cup \overline{E}$, with $v_1' \in V_1 \backslash \{v_1\}$. Clearly, both sets belong to $S_a(G)$ and therefore their incidence vectors satisfy $bx = \beta$. As $bx^F = \beta$, by (4.31) we get

$$b(v_1't_1) = b(t_1t_2) = b(v_1't_2). \tag{4.34}$$

From (4.29), (4.31) and (4.34), we have shown that

$$b(f) = \rho, \text{ for all } f \in [s, \{t_1, t_2\}] \cup [t_1, t_2] \cup [v_1, t_2] \cup [v_1', \{t_1, t_2\}],$$

for some scalar $\rho \in I\!R$. As $v_1'$ is an arbitrary node in $V_1 \backslash \{v_1\}$, it also follows that

$$b(f) = \rho, \text{ for all } f \in [s, \{t_1, t_2\}] \cup [t_1, t_2] \cup [v_1, t_2] \cup [V_1 \backslash \{v_1\}, \{t_1, t_2\}]. \tag{4.35}$$

If there are edges in $[v_1, t_1] \backslash \{e\}$, we still need to prove that their coefficient in $b$ is equal to $\rho$. Suppose this is the case and let $v_1t_1$ be an edge of $[v_1, t_1] \backslash \{e\}$. Then, by (iv), the edge set $\tilde{F} = \{v_1t_1, v_1t_2, st_2\} \cup \overline{E}$ is in $S(G)$. Clearly, its incidence vector $x^{\tilde{F}}$ satisfies $ax = \alpha$. Thus $bx^{\tilde{F}} = \beta$. As $bx^F = \beta$, by (4.35) we obtain

$$b(v_1t_1) = \rho. \tag{4.36}$$

Since $v_1t_1$ is an arbitrary edge of $[v_1, t_1] \backslash \{e\}$, this implies that

$$b(f) = \rho, \text{ for all } f \in [v_1, t_1] \backslash \{e\}. \tag{4.37}$$

From (4.28),(4.32),(4.35),(4.37), we have $b = \rho a$. As $bx \geq \beta$ is a facet-defining inequality different from a trivial one, by Lemma 4.5.2, it follows that $\rho > 0$. □

Note that triple path-cut inequalities never define facets of $P_G(D, L)$ when $L = 3$. It is indeed impossible to find a feasible solution satisfying such an inequality with equality while containing an edge in $[V_1, V_3]$ (of coefficient 2 in the triple path-cut).

## 4.6 Separations

In this section, we discuss the separation problem for the inequalities introduced before. Let $x^*$ be a vector in $\mathbb{R}^{|E|}$. In the sequel, we will denote by $G_{x^*} = (N, E_{x^*})$ the *support graph* of $x^*$, that is the graph induced by the edges $e$ such that $x^*(e) > 0$.

The $st$-cut constraints (4.1) can be separated exactly using the Gomory-Hu algorithm [29]. This produces the so-called Gomory-Hu tree, which has the property that for all pairs of nodes $s, t \in N$ the minimum $st$-cut in the tree is also a minimum $st$-cut in $G_{x^*}$. To do this, we use the algorithm developed by Gusfield [43]. This one requires $|N| - 1$ maximum flow computations. In practice, to speed up the computation, we first use a simple heuristic to try to quickly find a violated $st$-cut inequality. This one goes as follows. We iteratively contract the edges by decreasing $x^*$ values until either the total value of the shrunk graph $\tilde{G} = (\tilde{V}, \tilde{E})$ is less than its number of nodes, or there only remain two nodes. In the first case, at least one of the nodes of $\tilde{G}$ induces a cut violated by the restriction of $x^*$ on $\tilde{E}$. By expanding this node, we obtain a violated cut in $G_{x^*}$. In the second case, we check if the cut between the two nodes of $\tilde{G}$ is violated or not. Of course, in both cases, we verify if the cut obtained separates two nodes of the same demand. If this heuristic is unsuccessful, we then generate the Gomory-Hu tree, using the Gusfield algorithm, to separate the cut constraints exactly.

The $L$-path-cut inequalities (4.2) can also be separated in polynomial time when $L = 2, 3$. In fact, for a fixed demand $\{s, t\} \in D$, the separation problem reduces to finding a minimum weight edge subset that intersects all $L$-$st$-paths. This has been shown polynomially solvable in [25]. In practice, when $L = 2$, we do the following for each demand $\{s, t\} \in D$. We consider the partition $\Pi = (V_0, V_1, V_2, V_3)$ with $V_0 = \{s\}$, $V_3 = \{t\}$, and where $V_1$ and $V_2$ are constructed as follows. For each node $u \in N \backslash \{s, t\}$, we put $u$ in $V_1$ if $x(su) \geq x(ut)$, and $u$ in $V_2$ if not. We then test the violation of the corresponding 2-$st$-path-cut inequality. When $L = 3$, for each demand $\{s, t\} \in D$, we first check if the minimum weight edge set cutting all the 3-$st$-paths of $G_{x^*}$ is less than 2. If yes, there is a 3-$st$-path-cut inequality violated by $x^*$. That one is then build through a breadth first search from $s$ in $G_{x^*}$.

Let us turn now our attention to the separation problem for the rooted-partition inequalities (4.11). We shall prove that this problem for $L = 2$, when $p$ is odd, and the partition sets, except $V_0$, are singletons, can be solved in polynomial time. Note that these two conditions are necessary for the rooted-partition inequalities to be facet defining, see Theorem 4.5.4. As it will turn out, that problem will reduce to minimizing a submodular function on a parity subfamily of a lattice family.

Let $M$ be a finite set. A family $C$ of subsets of elements of $M$ is called a *lattice family* if

$$X \cup Y, X \cap Y \in C, \text{ for all } X, Y \in C.$$

A function $f : C \to I\!R$ is said to be *submodular* if

$$f(X \cup Y) + f(X \cap Y) \leq f(X) + f(Y), \text{ for all } X, Y \in C.$$

A subcollection $D$ of $C$ is called a *parity family* if

$$X \cap Y \in D \iff X \cup Y \in D, \text{ for all } X, Y \in C \backslash D.$$

In [28], Goemans and Ramakrishnan have shown the following result.

**Theorem 4.6.1.** [28] *Given a submodular function $f$ on a lattice family $C$, and a parity family $D$ of $C$, a set $U$ minimizing $f(U)$ over $U \in D$ can be found in polynomial time.*

**Theorem 4.6.2.** *The separation problem for the rooted-partition inequalities*

$$x(\Delta_\Pi(G)) \geq \lceil 3p/2 \rceil, \tag{4.38}$$

*when $L = 2$, $p$ odd, and $V_i = \{t_i\}$ for $i = 1, \ldots, p$, can be solved in polynomial time.*

*Proof.* Let $x \in I\!R^{|E|}$. As $p$ is odd, the separation problem for inequalities (4.38) is equivalent to

$$\min\, x(\Delta_\Pi(G)) - 3p/2 - 1/2, \tag{4.39}$$

over the partitions $\Pi = (V_0, V_1, \ldots, V_p)$ of $N$ such that $V_i = \{t_i\}$ for $i = 1, \ldots, p$. Problem (4.39) can also be written as

$$\min_{S \subseteq \bigcup_{i=1}^{d} \{t_i\},\, |S| \text{ odd}} x(E(S)) + x(\delta(S)) - 3|S|/2 - 1/2. \tag{4.40}$$

Now let $M$ be the set $N$ of nodes of $G$, $C$ be the set of subsets of $N$ contained in $\bigcup_{i=1}^{d}\{t_i\}$, $f(S) = x(E(S)) + x(\delta(S)) - 3|S|/2 - 1/2$, and $D = \{X \in C : |X| \text{ odd }\}$.

It is clear that $C$ is a lattice family and that $D$ is a parity family. Also it can be easily seen that $f$ is submodular on $C$. Consequently, by Theorem 4.6.1, the result follows. □

In our Branch-and-Cut algorithm, we will rather use a heuristic separation for the rooted-(opt-)partition inequalities (4.11) ((4.12)). This one works as follows for each source node $s$ with at least two destinations. We consider the nodes that are not destinations of $s$ as Steiner nodes. If $L = 2$, we first look for triangles formed by edges with value 1. If $\mathcal{C}$ is such a triangle, then we contract $\mathcal{C}$ into a pseudo-node $w$. If $\mathcal{C}$ contains $s$ (resp. a destination node but not $s$) (resp. only Steiner nodes), then $w$ will take the role of $s$ (resp. a destination node) (resp. a Steiner node) in the new graph. After this possible step, we consider the Steiner nodes. If a Steiner $u$ is adjacent to at least one destination node, then we contract the edge that has the highest value, between $u$ and a destination node. The new node is considered as a destination node. If not, then we contract $u$ and $s$ and consider the new node as $s$. At the end of this procedure, we get a graph without Steiner node, and containing, say $p$, destination nodes $t_1, \ldots, t_p$. This graph gives rise to a partition $V_0, V_1, \ldots, V_p$ of $N$ such that $s \in V_0$ and $V_i$ contains at least a destination node of $s$, for $i = 1, \ldots, p$. If $L = 3$, then we check whether the rooted-partition induced by this partition is violated. If $L = 2$, then we consider a rooted-opt-partition obtained from $V_0, V_1, \ldots, V_p$ as follows. We consider an edge $uv$ with maximum value such that $u \in V_i$ and $v \in V_j$ for some $i, j \in \{1, \ldots, p\}$, $i < j$, and at least one of the sets $V_i \backslash \{u\}$ and $V_j \backslash \{v\}$ contains at least a destination node. If $V_i \backslash \{u\}$ (resp. $V_j \backslash \{v\}$) contains a destination, then we consider the rooted-opt-partition induced by $e = uv$ together with the partition $\Pi' = (V_0', V_1', \ldots, V_{p+1}')$ given by

$$\begin{array}{lcll} V_l' & = & V_l & \text{for } l = 0, \ldots, i-1, \\ V_{l-1}' & = & V_l & \text{for } l = i+1, \ldots, j-1, \\ V_{l-2}' & = & V_l & \text{for } l = j+1, \ldots, p, \\ V_{p-1}' & = & V_j & \text{(resp. } V_i\text{)}, \\ V_p' & = & V_i \backslash \{u\} & \text{(resp. } V_j \backslash \{v\}\text{)}, \\ V_{p+1}' & = & \{u\} & \text{(resp. } \{v\}\text{)}. \end{array}$$

We notice that, as shown in the proof of Theorem 4.5.4 (ii), this rooted-opt-partition inequality is stronger than the rooted-partition one. We then test the violation of this rooted-opt-partition inequality. Moreover, in both cases ($L = 2$ or $L = 3$), if the tested inequality is not violated, we then contract two sets $V_i, V_j$ such that $[V_i, V_j]$ contains an edge having the biggest value in $\Delta_\Pi(G)$, and test this smaller partition. This procedure is stopped whenever the number of partition subsets becomes 2, since the corresponding rooted-partition is then an $st$-cut.

When $L = 2, 3$, we also separate heuristically the double cut inequalities (4.6)-(4.7) and triple path-cut inequalities (4.9)-(4.10). In what follows, we present their separation procedures for $L = 2$. Those for $L = 3$ are similar. For the former class, we apply the following for every demand $\{s_1, t_1\} \in D$ and every terminal node $t_2$ different from $s_1, t_1$. We consider the partition $\Pi = (V_0, \ldots, V_3)$ where $V_0 = \{s_1\}$, $V_1 = \{t_2\}$, $V_2 = N \backslash \{s_1, t_1, t_2\}$, $V_3 = \{t_1\}$. The idea behind this is to get a double cut inequality that, by Theorems 4.5.5-4.5.6, may define a facet of $P_G(D, 2)$. We select $e \in [V_1, V_2]$ having the biggest value and then test the violation of the double cut inequality corresponding to $\Pi$ and $e$. (We also test the partition obtained by exchanging the roles of $s_1$ and $t_1$.)

For the latter class, the separation procedure goes as follows for any source node $s$ with at least two destinations. We look for two destination nodes $t_1, t_2$ of $s$ such that the triangle $s, t_1, t_2$ has minimum $x^*$ value. We then consider the partition $\Pi = (V_0, V_1, \ldots, V_4)$ such that $V_0 = \{s\}$, $V_1 = N \backslash \{s, t_1, t_2\}$, $V_2 = \emptyset$, $V_3 = \{t_1\}$, $V_4 = \{t_2\}$. Note that this partition satisfies the necessary conditions (i), (ii), (iii) of Theorem 4.5.8. Also, since we consider graphs with at least five nodes, we have that $|V_1| \geq 2$. For every node $u$ of $V_1$, if $x^*(su) \leq 1/2(x^*(ut_1) + x^*(ut_2))$, then we move $u$ from $V_1$ to $V_2$, and consider the new partition, still denoted by $\Pi$. The motivation behind this is, as before, to reduce as much as possible the lefthand side of the generated inequality. This process is stopped if $V_1$ has only one node left. If, after this step, $V_2$ is still empty, then we take a node $u$ from $V_1$ such that $x^*(su) = \min_{v \in V_1}\{x^*(sv)\}$, and we transfer it to $V_2$. We then test if the triple path-cut inequality corresponding to $\Pi$ is violated. The edge $e$ to be removed is chosen as the edge with the maximum $x^*$ value between $V_1$ and $\{t_1, t_2\}$. If $e$ is incident to $t_2$, then we exchange the roles of $t_1$ and $t_2$ for $e$ to belong to $[V_1, V_3]$.

## 4.7 Branch-and-Cut and computational results

Based upon the previous theoretical results, we have developed a Branch-and-Cut algorithm in order to efficiently solve the THNDP when $L = 2, 3$. The algorithm has been implemented in C++, using BCP to manage the branching tree and CPLEX 8.11 as linear solver, and tested on a Pentium III at 933 MHz with 384 Mo of RAM under Linux. The maximum runtime has been fixed to 5 hours. The results presented here essentially concern the case where the demands are rooted in a node $s$, since we have focussed our polyhedral study on this case. Nevertheless, our algorithm is adapted to any set of demands, and we also present some computational results in that case. For each instance tested, we have run the algorithm twice, once with the constraints (4.1)-(4.4) only, and a second time with also inequalities (4.5)-(4.12) depending on their respective validity for $L = 2, 3$.

To begin the optimization, we consider the linear program consisting of the cut inequalities associated to the demand nodes, and the trivial inequalities. Moreover, in the second run of the algorithm, for each source node $s$ with at least two destinations, we add to this basic program the rooted-partition inequality where each destination of $s$ corresponds to an element of the partition and all the other nodes are put in $V_0$.

In the Branch-and-Cut algorithm, we have to check whether or not an optimal solution of a relaxation to the THNDP is feasible. An optimal solution $x^*$ of a relaxation is feasible for the THNDP if it is an integer vector satisfying the $st$-cut and $L$-$st$-path-cut inequalities. Verifying if an integer solution $x^*$ is feasible for the THNDP can be done in an efficient way. For every edge $e$ of $G_{x^*}$ and every demand $\{s, t\} \in D$, we check if the shortest $st$-path in $G_{x^*} - e$ is of length $\leq L$. If this the case, then by Lemma 4.3.1 $x^*$ is feasible. If not, this means there is no $L$-$st$-path not containing $e$. This implies that $x^*$ is not feasible.

Another important issue in the effectiveness of the Branch-and-Cut algorithm is to compute a good upper bound. For this, in the solution of the current linear program, we first round up to 1 all the variables with a value $\geq 0.3$ and round down to 0 those with value $< 0.3$. We try to improve this solution by deleting all the edges incident to a Steiner node whose total value is less than 2. We then verify if the resulting integer solution is feasible. If not, we apply the same procedure for the solution obtained by rounding up to 1 all the fractional values.

If an optimal solution $x^*$ of the linear relaxation of the THNDP is not feasible, the algorithm generates additional valid inequalities of $P_G(D, L)$ violated by $x^*$. Their separation is realized in the following order :

- $st$-cut inequalities,
- $L$-path-cut inequalities,
- double cut inequalities,
- rooted-(opt-)partition inequalities (if $L = 2$),
- triple path-cut inequalities.

We remark that all inequalities are global (i.e. valid at every node of the Branch-and-Cut tree) and several of them can be added at each iteration. Moreover, we go to the next class only if we do not find any violated inequalities in the current class.

To separate the different inequalities, we use the algorithms described in Section 4.6. All our separation procedures are applied on the support graph of $x^*$, that is $G_{x^*}$, where $x^*$ is the solution of the current relaxation. When solving instances of the THNDP, we remarked that the exact separation of the $st$-cut inequalities is time consuming. Therefore, we decided to perform this exact separation after that of the $L$-path-cut inequalities.

To store the generated inequalities, we use a pool whose size increases dynamically. Inequalities in the pool can be removed from the current linear program when they are not active. Also, they are the first inequalities to be separated. If all the inequalities in the pool are satisfied by the current solution, we then separate the classes of inequalities according to the order given above.

The computational results presented here concern randomly generated instances and instances coming from real applications. The instances consist in complete graphs with edge costs equal to rounded Euclidean distances. The tests were performed for $L = 2, 3$. In practice, note that the bound on the routing paths does not usually exceed 4. The second set of instances comes from the network of the Belgian telecommunications operator, Belgacom, on 52 cities and subsets of these. The random problems were generated with $n = 10$ to $n = 40$ nodes, with different number $d$ of demands. For each couple $(n, d)$, five instances were tested.

In the various tables, the entries are:

| | | |
|---|---|---|
| $n$ | : | the number of nodes of the problem, |
| $d$ | : | the number of demands, |
| Cu | : | the number of generated $st$-cut inequalities (run 2), |
| Pc | : | the number of generated $L$-path-cut inequalities (run 2), |
| Dc | : | the number of generated double cut inequalities (run 2), |
| Ro | : | the number of generated rooted-(opt-)partition (if $L = 2$) inequalities (run 2), |
| Tp | : | the number of generated triple path-cut inequalities (run 2), |
| o1 | : | the number of problems solved to optimality (run 1) over five instances tested, |
| o2 | : | the number of problems solved to optimality (run 2) over five instances tested, |
| Gap1 | : | the gap between the best upper bound and the lower bound obtained at the root node of the Branch-and-Cut tree in the first run, |
| Gap2 | : | the gap between the best upper bound and the lower bound obtained at the root node of the Branch-and-Cut tree in the second run, |
| Gt1 | : | the gap between the best upper bound and the best lower bound obtained in the first run, |
| Gt2 | : | the gap between the best upper bound and the best lower bound obtained in the second run, |
| Tree1 | : | the number of nodes in the Branch-and-Cut tree for the first run, |
| Tree2 | : | the number of nodes in the Branch-and-Cut tree for the second run, |
| CPU1 | : | the total time of the first run in seconds, |
| CPU2 | : | the total time of the second run in seconds. |

The first two tables report the average results for the random instances, obtained in the case of rooted demands, for $L = 2$ and $L = 3$, respectively.

In Table 4.1, we remark that, up to 20 nodes and 15 demands, all problems have been solved to optimality within the time limit. Besides one exception, this is also the case for 30 (resp. 40) nodes when there are 15 (resp. 10) destinations or less. When we consider 20 demands or more, only one instance has been solved in less than 5 hours. We can remark that not many $st$-cut inequalities are obtained for the different instances. This, in fact, is because, when $L = 2$, the $L$-$st$-path-cut inequalities dominate the $st$-cut constraints induced by non singletons (see Section 2.6). Note that,

Table 4.1: Results for random instances for $L = 2$ and rooted demands

| $n$ | $d$ | Cu | Pc | Dc | Ro | Tp | o1 | o2 | Gap1 | Gap2 | Gt1 | Gt2 | Tree1 | Tree2 | CPU1 | CPU2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 2 | 0 | 4 | 2 | 2 | 2 | 5 | 5 | 0.0 | 0.0 | 0.0 | 0.0 | 1 | 1 | 0.01 | 0.01 |
| 10 | 5 | 1 | 24 | 5 | 9 | 5 | 5 | 5 | 8.0 | 1.1 | 0.0 | 0.0 | 17 | 5 | 0.10 | 0.06 |
| 10 | 8 | 3 | 70 | 20 | 26 | 8 | 5 | 5 | 12.0 | 5.0 | 0.0 | 0.0 | 156 | 34 | 1.30 | 0.54 |
| 20 | 5 | 0 | 36 | 5 | 10 | 10 | 5 | 5 | 6.6 | 0.9 | 0.0 | 0.0 | 10 | 2 | 0.12 | 0.09 |
| 20 | 10 | 5 | 480 | 65 | 311 | 25 | 5 | 5 | 13.5 | 6.8 | 0.0 | 0.0 | 1143 | 547 | 32.02 | 40.88 |
| 20 | 15 | 13 | 3077 | 371 | 973 | 59 | 5 | 5 | 15.1 | 9.2 | 0.0 | 0.0 | 79054 | 35770 | 6711.73 | 7081.31 |
| 30 | 8 | 2 | 249 | 22 | 273 | 26 | 5 | 5 | 8.9 | 3.3 | 0.0 | 0.0 | 152 | 51 | 5.05 | 15.19 |
| 30 | 15 | 8 | 2465 | 207 | 1491 | 82 | 4 | 4 | 12.8 | 7.4 | 0.2 | 0.7 | 28501 | 7694 | 5023.44 | 5215.93 |
| 30 | 22 | 17 | 2982 | 327 | 1554 | 40 | 1 | 1 | 30.5 | 20.2 | 22.6 | 15.7 | 73564 | 26235 | 14591.42 | 14432.96 |
| 40 | 10 | 3 | 1714 | 87 | 2972 | 84 | 4 | 4 | 13.2 | 7.4 | 0.1 | 1.0 | 10458 | 2139 | 3696.43 | 4426.79 |
| 40 | 20 | 13 | 3252 | 213 | 3379 | 56 | 0 | 0 | 33.4 | 24.2 | 25.6 | 20.7 | 60744 | 10978 | 18000.00 | 18000.00 |
| 40 | 30 | 26 | 3191 | 234 | 1626 | 19 | 0 | 0 | 43.3 | 36.3 | 39.1 | 34.1 | 53753 | 12859 | 18000.00 | 18000.00 |

in the second run, a significant number of double cut, rooted-opt-partition, and triple path-cut inequalities have been generated. If these yield little impact on the number of instances solved to optimality, or on the CPU time, we remark that the gaps (Gap2, Gt2) and, in particular, the size of the Branch-and-Cut tree (Tree2) are significantly reduced with respect to Gap1, Gt1, and Tree1. We may also observe that for the instances with 20 demands or more, the gap at the root node in the second run (Gap2) is better than the final gap obtained after 5 hours in the first run (Gt1).

Table 4.2 gives the computational results for the same instances as those of Table 4.1, when $L = 3$.

Table 4.2: Results for random instances for $L = 3$ and rooted demands

| $n$ | $d$ | Cu | Pc | Dc | Ro | Tp | o1 | o2 | Gap1 | Gap2 | Gt1 | Gt2 | Tree1 | Tree2 | CPU1 | CPU2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 2 | 5 | 0 | 2 | 3 | 1 | 5 | 5 | 0.0 | 0.0 | 0.0 | 0.0 | 1 | 1 | 0.01 | 0.01 |
| 10 | 5 | 11 | 53 | 10 | 14 | 3 | 5 | 5 | 11.3 | 5.6 | 0.0 | 0.0 | 58 | 17 | 0.70 | 0.21 |
| 10 | 8 | 9 | 198 | 20 | 8 | 2 | 5 | 5 | 12.3 | 6.4 | 0.0 | 0.0 | 278 | 89 | 7.26 | 1.50 |
| 20 | 5 | 28 | 146 | 13 | 45 | 2 | 5 | 5 | 8.3 | 5.2 | 0.0 | 0.0 | 37 | 17 | 2.25 | 1.35 |
| 20 | 10 | 30 | 3407 | 52 | 312 | 9 | 5 | 5 | 13.9 | 6.8 | 0.0 | 0.0 | 3681 | 1247 | 2403.97 | 437.00 |
| 20 | 15 | 38 | 24953 | 170 | 422 | 21 | 0 | 0 | 30.6 | 16.1 | 21.8 | 8.6 | 10049 | 11971 | 18000.00 | 18000.00 |
| 30 | 8 | 30 | 2250 | 42 | 384 | 1 | 5 | 5 | 10.3 | 5.3 | 0.0 | 0.0 | 996 | 336 | 1927.48 | 468.81 |
| 30 | 15 | 60 | 18509 | 122 | 1071 | 5 | 0 | 1 | 29.9 | 16.4 | 21.9 | 9.9 | 5326 | 5087 | 18000.00 | 16647.14 |
| 30 | 22 | 39 | 15703 | 105 | 318 | 4 | 0 | 0 | 42.8 | 30.4 | 38.5 | 27.7 | 3859 | 5753 | 18000.00 | 18000.00 |
| 40 | 10 | 58 | 9351 | 71 | 1087 | 4 | 2 | 3 | 16.7 | 7.4 | 6.6 | 1.7 | 1965 | 1099 | 11925.39 | 8710.00 |
| 40 | 20 | 61 | 11442 | 76 | 940 | 1 | 0 | 0 | 46.0 | 36.7 | 42.4 | 34.7 | 1808 | 1930 | 18000.00 | 18000.00 |
| 40 | 30 | 49 | 11344 | 85 | 274 | 4 | 0 | 0 | 57.3 | 45.7 | 55.1 | 44.4 | 1525 | 1547 | 18000.00 | 18000.00 |

Similar observations as for Table 4.1 can be made for Table 4.2. However, the improvement between the first and second runs is even more important. Indeed, one can remark that, for the instances with $(n, d) = (10, 8)$, $(20, 10)$ or $(30, 8)$, the CPU time is almost divided by 5. This can be explained by the fact that the $st$-cut inequalities when $L = 2$ have been all generated by our heuristic, while, for $L = 3$, the exact separation, which takes more time, has been used. For this reason, it is natural to come with more exciting results in run 1 when $L = 2$ than in run 1 when $L = 3$. Moreover, for the instances with $(n, d) = (30, 15)$ or $(40, 10)$, we can see that one more instance has been solved to optimality in run 2. And finally, the gap Gt2 has been reduced for the big instances by almost 10% with respect to Gt1.

Tables 4.3 and 4.4 summarize the average results obtained for the same instances, but with arbitrary sets of demands.

Table 4.3: Results for random instances for $L = 2$ and arbitrary demands

| $n$ | $d$ | Cu | Pc | Dc | Ro | Tp | o1 | o2 | Gap1 | Gap2 | Gt1 | Gt2 | Tree1 | Tree2 | CPU1 | CPU2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 5 | 3 | 59 | 21 | 10 | 20 | 5 | 5 | 10.2 | 8.8 | 0.0 | 0.0 | 21 | 21 | 0.16 | 0.24 |
| 20 | 5 | 3 | 76 | 14 | 7 | 20 | 5 | 5 | 4.7 | 3.8 | 0.0 | 0.0 | 7 | 9 | 0.22 | 0.40 |
| 20 | 10 | 7 | 761 | 52 | 20 | 102 | 5 | 5 | 8.3 | 7.8 | 0.0 | 0.0 | 625 | 358 | 50.58 | 45.19 |
| 30 | 8 | 5 | 650 | 66 | 56 | 154 | 5 | 5 | 8.5 | 8.5 | 0.0 | 0.0 | 459 | 489 | 28.61 | 43.40 |
| 30 | 15 | 13 | 5397 | 412 | 119 | 910 | 2 | 2 | 12.5 | 12.7 | 3.9 | 4.6 | 23291 | 15357 | 11748.40 | 11613.36 |
| 40 | 10 | 6 | 519 | 50 | 15 | 73 | 5 | 5 | 5.4 | 5.0 | 0.0 | 0.0 | 107 | 63 | 15.28 | 16.44 |
| 40 | 20 | 13 | 6565 | 363 | 15 | 736 | 0 | 0 | 17.0 | 19.3 | 11.4 | 14.0 | 10859 | 8349 | 18000.00 | 18000.00 |

Table 4.4: Results for random instances for $L = 3$ and arbitrary demands

| $n$ | $d$ | Cu | Pc | Dc | Ro | Tp | o1 | o2 | Gap1 | Gap2 | Gt1 | Gt2 | Tree1 | Tree2 | CPU1 | CPU2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 5 | 9 | 68 | 21 | 4 | 2 | 5 | 5 | 7.0 | 5.8 | 0.0 | 0.0 | 29 | 22 | 0.35 | 0.34 |
| 20 | 5 | 20 | 565 | 46 | 6 | 0 | 5 | 5 | 8.5 | 7.7 | 0.0 | 0.0 | 103 | 91 | 12.12 | 10.01 |
| 20 | 10 | 31 | 20280 | 395 | 7 | 3 | 2 | 2 | 20.7 | 18.5 | 8.0 | 6.7 | 5042 | 4369 | 12555.41 | 11183.45 |
| 30 | 8 | 50 | 14251 | 282 | 40 | 4 | 3 | 3 | 15.7 | 14.0 | 3.1 | 2.1 | 1948 | 1816 | 8489.24 | 8181.75 |
| 30 | 15 | 29 | 19982 | 318 | 4 | 1 | 0 | 0 | 56.7 | 56.4 | 53.7 | 53.5 | 1126 | 1190 | 18000.00 | 18000.00 |
| 40 | 10 | 64 | 16501 | 279 | 9 | 1 | 0 | 0 | 30.1 | 33.5 | 23.8 | 28.2 | 1605 | 1709 | 18000.00 | 18000.00 |
| 40 | 20 | 29 | 12519 | 148 | 4 | 1 | 0 | 0 | 61.5 | 61.4 | 60.2 | 60.0 | 274 | 301 | 18000.00 | 18000.00 |

Unfortunately, for Tables 4.3 and 4.4, no significant improvement between the first and second runs is observed. We believe that this is because the valid inequalities presented in this chapter are more adapted to rooted demands. However, many instances in both cases have been solved to optimality. For $L = 2$, all the instances with up to 10 demands were solved to optimality. For $L = 3$, this was the case for up to 5 demands only.

In Tables 4.5 and 4.6, we report the computational results obtained for real instances for $L = 2$ and $L = 3$, respectively. Here, the instances with 10 demands are rooted, while the others have multiples sources.

Table 4.5: Results for real instances when $L = 2$

| $n$ | $d$ | Cu | Pc | Dc | Ro | Tp | Gap1 | Gap2 | Gt1 | Gt2 | Tree1 | Tree2 | CPU1 | CPU2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 11 | 10 | 0 | 37 | 25 | 28 | 3 | 9.9 | 3.3 | 0.0 | 0.0 | 201 | 29 | 1.83 | 0.59 |
| 11 | 30 | 0 | 793 | 4 | 48 | 49 | 17.3 | 8.9 | 0.0 | 0.0 | 135 | 95 | 7.86 | 10.67 |
| 11 | 55 | 1 | 2376 | 8 | 51 | 51 | 20.9 | 10.1 | 0.0 | 0.0 | 245 | 235 | 118.30 | 88.88 |
| 30 | 10 | 4 | 241 | 35 | 302 | 27 | 9.9 | 3.4 | 0.0 | 0.0 | 215 | 101 | 5.19 | 18.58 |
| 30 | 30 | 2 | 1862 | 7 | 347 | 67 | 17.3 | 8.9 | 0.0 | 0.0 | 103 | 87 | 35.29 | 94.78 |
| 30 | 55 | 0 | 8767 | 4 | 610 | 98 | 20.9 | 10.1 | 0.0 | 0.0 | 227 | 321 | 463.32 | 859.75 |
| 52 | 10 | 1 | 613 | 48 | 1041 | 59 | 9.9 | 4.5 | 0.0 | 0.0 | 641 | 315 | 35.30 | 226.05 |
| 52 | 30 | 0 | 4184 | 13 | 1239 | 156 | 17.3 | 9.2 | 0.0 | 0.0 | 161 | 235 | 241.0 | 1130.67 |
| 52 | 55 | 0 | 13540 | 3 | 6920 | 90 | 20.9 | 11.1 | 0.0 | 7.4 | 221 | 1051 | 3119.40 | 18000.00 |

Table 4.6: Results for real instances when $L = 3$

| $n$ | $d$ | Cu | Pc | Dc | Ro | Tp | Gap1 | Gap2 | Gt1 | Gt2 | Tree1 | Tree2 | CPU1 | CPU2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 11 | 10 | 10 | 135 | 19 | 3 | 0 | 11.2 | 2.4 | 0.0 | 0.0 | 869 | 81 | 21.82 | 1.13 |
| 11 | 30 | 11 | 2887 | 234 | 14 | 5 | 16.2 | 8.8 | 0.0 | 0.0 | 4627 | 891 | 1655.84 | 126.39 |
| 11 | 55 | 11 | 3485 | 351 | 9 | 0 | 16.1 | 7.3 | 0.0 | 0.0 | 3767 | 721 | 2741.18 | 196.13 |
| 30 | 10 | 3 | 43 | 3194 | 50 | 339 | 11.2 | 3.8 | 0.0 | 0.0 | 3235 | 603 | 3678.49 | 348.73 |
| 30 | 30 | 14 | 24111 | 423 | 659 | 44 | 17.1 | 8.9 | 5.3 | 1.8 | 2233 | 3115 | 18000.00 | 18000.00 |
| 30 | 55 | 16 | 25376 | 512 | 576 | 45 | 53.2 | 7.6 | 46.8 | 1.0 | 1605 | 2361 | 18000.00 | 18000.00 |
| 52 | 10 | 51 | 14309 | 68 | 1328 | 9 | 11.3 | 7.3 | 3.2 | 2.6 | 1643 | 2221 | 18000.00 | 18000.00 |
| 52 | 30 | 31 | 14685 | 80 | 535 | 4 | 21.5 | 12.2 | 13.9 | 6.6 | 449 | 407 | 18000.00 | 18000.00 |
| 52 | 55 | 42 | 11950 | 103 | 1234 | 11 | 53.5 | 13.7 | 49.1 | 10.6 | 383 | 387 | 18000.00 | 18000.00 |

Consider first Table 4.6. We can remark that, for the instances solved to optimality, the CPU time in the first run has been divided by more than 10 in the second run. And for the other instances, the final gap Gt2 has been considerably reduced compared to Gt1. The most significant case is the instance with 30 nodes and 55 demands. The final gap decreases from 46.8% to 1%. However, quite surprisingly, in Table 4.5, for the same instances with $L = 2$, the second run does not permit to improve neither the CPU time, nor the final gap. Only the gap at the root node decreases between the two runs.

Here below, we present the best solutions obtained for the real instances with 52 Belgian cities, for $L = 2$ and $L = 3$, respectively. The case where $d = 10$ (Figure 4.7) corresponds to a rooted situation where we would like to link the "capitals" of the ten regions of Belgium to Brussels (BRU). When $d = 30$ (see Figure 4.8), we split these ten cities into two groups, namely BRG-GEN-ANT-HAS-LEU and WAV-NAM-LIE-MON-ARL, and, in each group plus Brussels, we require two $L$-paths relatively to all the possible pairs. Finally, when $d = 55$ (Figure 4.9), we consider all demands based on these ten cities plus Brussels.

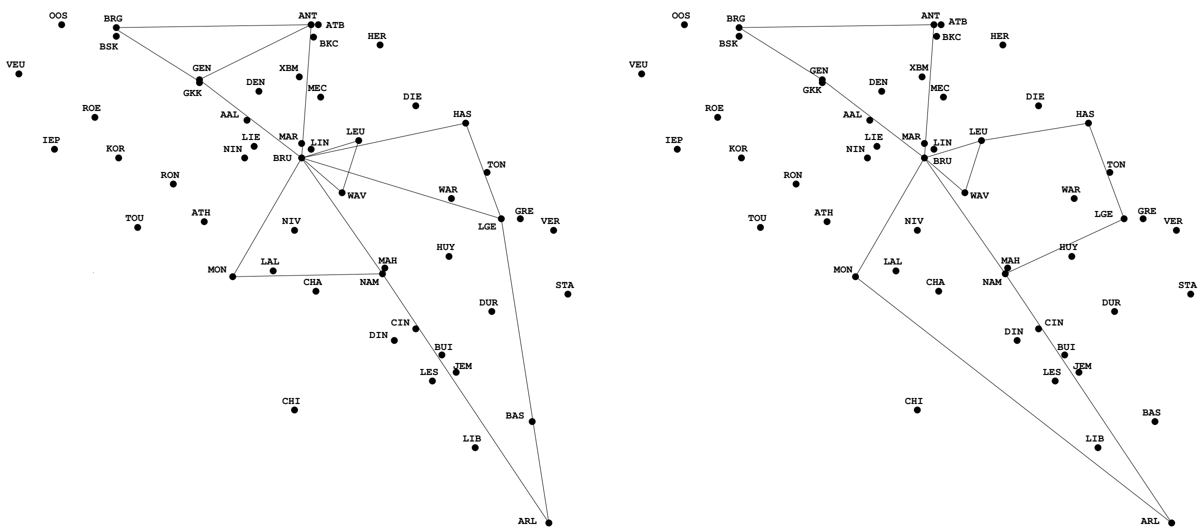


Figure 4.7: Best solutions found for 52 cities with 10 demands, on the left, when $L = 2$ (optimal), and, on the right, when $L = 3$

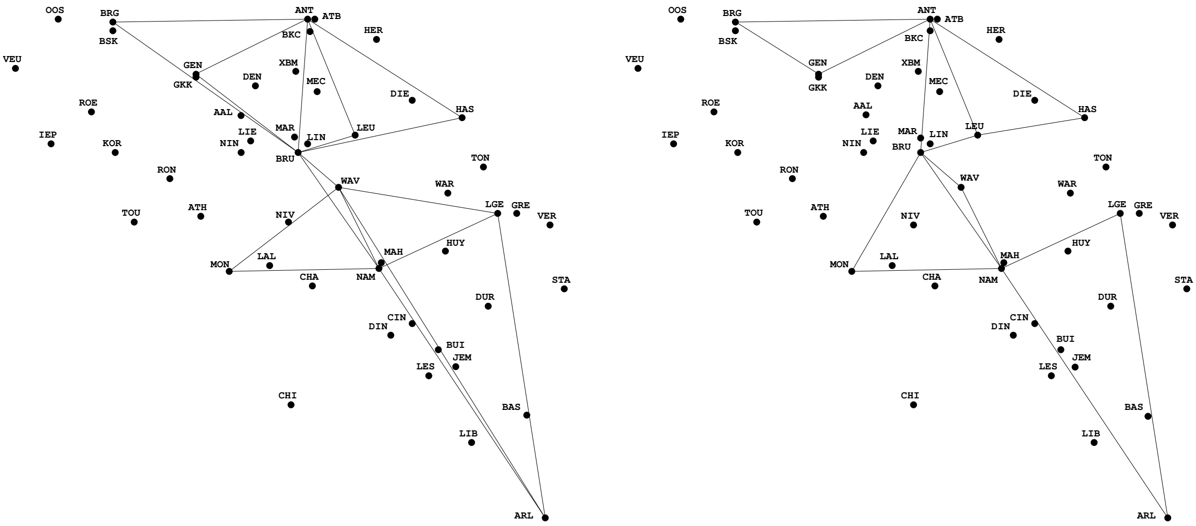


Figure 4.8: Best solutions found for 52 cities with 30 demands, on the left, when $L = 2$ (optimal), and, on the right, when $L = 3$

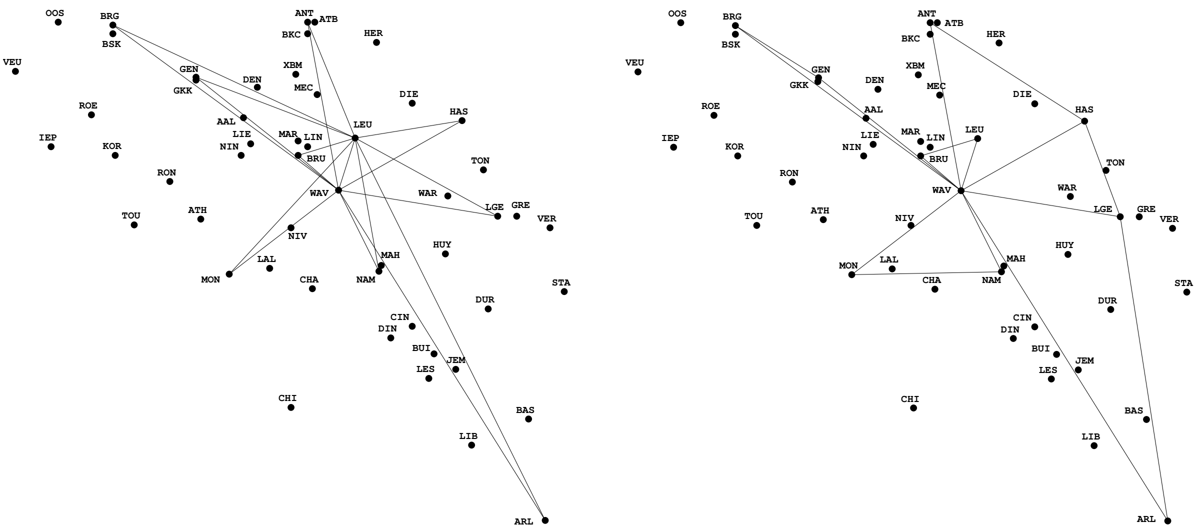


Figure 4.9: Best solutions found for 52 cities with 55 demands, on the left, when $L = 2$ (optimal), and, on the right, when $L = 3$

## 4.8 Concluding remarks

In this chapter, we have studied the Two-edge connected Hop-constrained Network Design Problem (THNDP). We have proved that the problem remains strongly $NP$-hard even in the rooted case and for any fixed $L \geq 2$. We have given an integer programming formulation of the THNDP when $L = 2, 3$, and described various families of valid inequalities. We have then focussed on the rooted case and $L = 2, 3$, where we have studied necessary conditions and sufficient conditions for these inequalities to be facet-defining. We have also discussed separation routines for the different classes of inequalities. In particular, for the rooted-partition inequalities, when the elements of the partition, different from the one containing the source node, are singletons, and when there are an odd number of them, we have shown that the associated separation problem can be reduced to the minimization of a submodular function, and hence, can be solved in polynomial time.

Using our polyhedral results, we have devised a Branch-and-Cut algorithm for $L = 2, 3$, and presented extensive computational results. We could estimate the effect of the double cut, the rooted-(opt-)partition and the triple path-cut inequalities in the Branch-and-Cut algorithm. This is particularly strong for the rooted case since these inequalities are more adapted to this case. For arbitrary demands, the best results have been realized for the real instances when $L = 3$. We could also measure the performance of our separation techniques.

Therefore, it would be of great interest to know additional classes of valid inequalities for the THNDP when the demand set contains multiple sources. Those could then be used as cutting planes in our Branch-and-Cut algorithm in order to improve the previous computational results in that case. The general THNDP (i.e. non necessarily rooted) will be considered in the next chapter.

# Chapter 5

# General Two-Edge Connected Hop-Constrained Network Design Problem

We continue here our study of the Two-edge connected Hop-constrained Network Design Problem, or THNDP. In Chapter 4, we already gave several results for this problem. However, even if the introduced classes of valid inequalities were given in all generality, they happened, by their structure, to be more effective in the case of rooted demands. Therefore, we will now present additional classes of valid inequalities, which rather take into account the interaction between node-disjoint demands. We will then see if they are useful in practice, when we are in presence of arbitrary demands (non necessarily rooted). This work is issued from the same collaboration as in the previous chapter, but has not been the object of a publication up to now.

## 5.1 Introduction

We do not recall here the statement of the THNDP problem and its related definitions. For this, the reader is referred to Section 4.1.

The chapter is organized as follows. In the following section, we present several new classes of valid inequalities for the THNDP when $L = 2$. While the inequalities given in Chapter 4 were based on several rooted demands, these ones are build on disjoint demands. In Section 5.3, we also propose a similar family of valid inequalities for the problem when $L = 3$. We then give heuristic separation procedures for these inequalities in Section 5.4, and test their effectiveness within the framework of our Branch-and-Cut algorithm in Section 5.5. We finally present some concluding remarks about this study in the last section.

## 5.2 New classes of valid inequalities for $L = 2$

In this section, we give new classes of valid inequalities for the THNDP when $L = 2$.

**Theorem 5.2.1.** *Let $V_0, V_1, \ldots, V_4$ be a partition of $N$, $d_1 = \{s_1, t_1\}$ be a demand of $D$ such that $|d_1 \cap V_0| = 1$ and $|d_1 \cap V_4| = 1$, and $V_1$ induce an $s_i t_i$-cut for some $i \in \{2, \ldots, d\}$. Let $e \in [V_1, V_2 \cup V_3]$. See Figure 5.1. The following inequality, called* a double path-cut inequality,

$$x([V_0, V_1]) + x([V_0, V_3]) + x([V_0 \cup V_1 \cup V_2, V_4]) + x([V_1, V_2 \cup V_3] \backslash \{e\}) \geq 3 \qquad (5.1)$$

*is valid for $P_G(D, 2)$.*

*Proof.* Inequality (5.1) can be obtained as the rounded-up half-sum of the following constraints:

- $s_i t_i$-cut inequality on subset $V_1$,
- 2-$s_1 t_1$-path-cut inequality on the partition $V_0, V_1 \cup V_2, V_3, V_4$,
- 2-$s_1 t_1$-path-cut inequality on the partition $V_0, V_2, V_1 \cup V_3, V_4$,
- trivial inequality $-x(e) \geq -1$,
- trivial inequalities $x(f) \geq 0$ for all $f \in [V_1, V_2 \cup V_3] \backslash \{e\}$.

□

The second class is similar to the first one, but it is based on 2-path-cut inequalities for different demands.

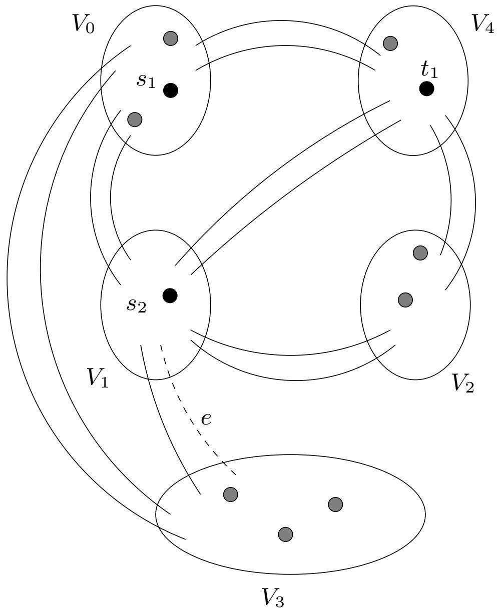


Figure 5.1: Support graph of a double path-cut inequality when $L = 2$ ($t_2 \in N \backslash V_1$)

**Theorem 5.2.2.** *Let $V_0, V_1, \ldots, V_4$ be a partition of $N$, and $d_1 = \{s_1, t_1\}$, $d_2 = \{s_2, t_2\}$ be two disjoint demands of $D$ such that $|d_1 \cap V_0| = 1$, $|d_1 \cap V_4| = 1$, $|d_2 \cap V_1| = 1$ and $|d_2 \cap V_2| = 1$. Let $e \in [V_1, V_2 \cup V_3]$. See Figure 5.2. The following inequality, called an* opposed double path-cut inequality*,*

$$x(\delta(V_0)) + x([V_2, V_4]) + x([V_1, V_2 \cup V_3] \backslash \{e\}) \geq 3 \tag{5.2}$$

*is valid for $P_G(D, 2)$.*

*Proof.* Inequality (5.2) can be obtained as the rounded-up half-sum of the following constraints:

- $s_1t_1$-cut inequality on subset $V_0$,
- 2-$s_1t_1$-path-cut inequality on $V_0, V_2, V_1 \cup V_3, V_4$,
- 2-$s_2t_2$-path-cut inequality on $V_1, V_0 \cup V_4, V_3, V_2$,
- trivial inequality $-x(e) \geq -1$,
- trivial inequalities $x(f) \geq 0$ for all $f \in [V_1, V_2 \cup V_3] \backslash \{e\}$.

□

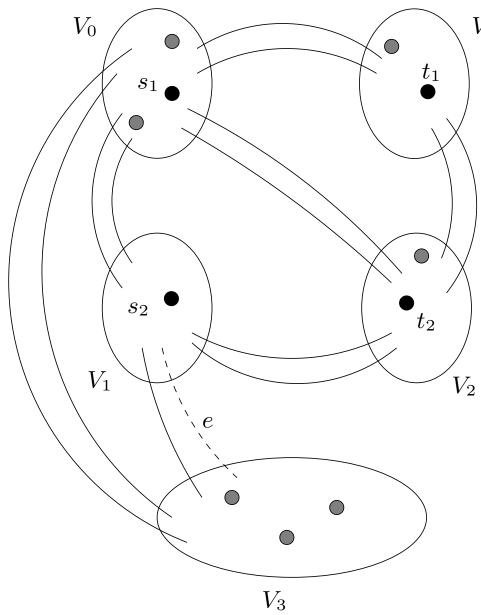


Figure 5.2: Support graph of an opposed double path-cut inequality

The following two classes are obtained as Chvátal-Gomory cuts of the two previous ones.

**Theorem 5.2.3.** *Let* $V_0, \ldots, V_4$ *be a partition of* $N$*, and* $d_1 = \{s_1, t_1\}$*,* $d_2 = \{s_2, t_2\}$ *be two disjoint demands such that* $|d_1 \cap V_0| = 1$*,* $|d_1 \cap V_4| = 1$*,* $|d_2 \cap V_1| = 1$*,* $|d_2 \cap V_2| = 1$*. Let* $e \in [V_1, V_2 \cup V_3]$ *and* $f \in [V_0, V_3 \cup V_4]$*. See Figure 5.3. The following inequality, called a* type I inequality*,*

$$x([V_0 \cup V_4, V_1 \cup V_2]) + x([V_0, V_3 \cup V_4] \backslash \{f\}) + 2x([V_1, V_2 \cup V_3] \backslash \{e\}) + x(e) \geq 4 \quad (5.3)$$

*is valid for* $P_G(D, 2)$*.*

*Proof.* Inequality (5.3) can be obtained as the rounded-up half-sum of the following constraints:

- double path-cut inequality associated to partition $V_1, V_0, V_4, V_3, V_2$, demands $d_2, d_1$, and edge $f$,
- opposed double path-cut associated to partition $V_1, V_0, V_4, V_3, V_2$, demands $d_2, d_1$, and edge $f$,
- 2-$s_2t_2$-path-cut inequality on partition $V_1, V_0, V_3 \cup V_4, V_2$,

- trivial inequality $-x(e) \geq -1$,

- trivial inequalities $x(g) \geq 0$ for all $g \in [V_1, V_2 \cup V_3] \backslash \{e\}$.

□

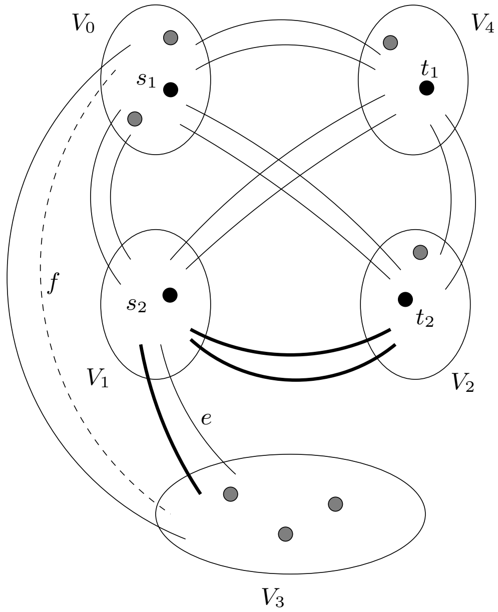


Figure 5.3: Support graph of a type I inequality

**Theorem 5.2.4.** *Let $V_0, \ldots, V_4$ be a partition of $N$, and $d_1 = \{s_1, t_1\}$, $d_2 = \{s_2, t_2\}$ be two disjoint demands of $D$ such that $|d_1 \cap V_0| = 1$, $|d_1 \cap V_4| = 1$, $|d_2 \cap V_1| = 1$, $|d_2 \cap V_2| = 1$. Let $e \in [V_1, V_2 \cup V_3]$. See Figure 5.4. The following inequality, called a* type II inequality,

$$x(\delta(V_0)) + x([V_1, V_4]) + x([V_2, V_4]) + 2x([V_1, V_2 \cup V_3] \backslash \{e\}) \geq 4 \tag{5.4}$$

*is valid for $P_G(D, 2)$.*

*Proof.* Inequality (5.4) can be obtained as the rounded-up half-sum of the following constraints:

- double path-cut inequality associated to partition $V_0, \ldots, V_4$, demands $d_1, d_2$, and edge $e$,

- opposed double path-cut inequality associated to partition $V_0, \dots, V_4$, demands $d_1, d_2$, and edge $e$,
- 2-$s_2t_2$-path-cut inequality on partition $V_1, V_0, V_3 \cup V_4, V_2$,
- trivial inequality $-x(e) \geq -1$,
- trivial inequalities $x(f) \geq 0$ for all $f \in [V_1, V_2 \cup V_3] \backslash \{e\}$.

□

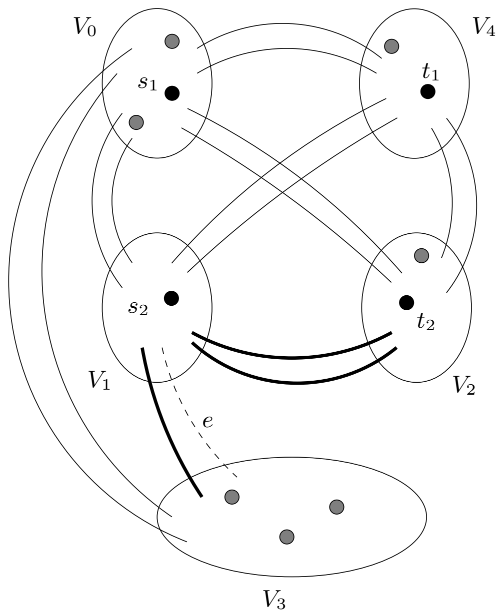


Figure 5.4: Support graph of a type II inequality

Our last class of valid inequalities gives the minimum number of edges in a multicut based on several disjoint demands.

**Theorem 5.2.5.** *Let* $\{s_1, t_1\}, \dots, \{s_{2p+1}, t_{2p+1}\}$, $1 \leq 2p+1 \leq d$, *be disjoint demands of* $D$, *and* $V_0, V_1, \dots, V_{4p+2}$ *be a partition of* $N$. *Let* $V_{2j-1} = \{s_j\}$ *and* $V_{2j} = \{t_j\}$ *for* $j = 1, \dots, 2p+1$, *and* $V_0 = N \backslash (\bigcup_{i=1,\dots,4p+2} V_i)$. *Suppose that* $|[s_i, t_i]| \leq 1$ *for* $i = 1, \dots, 2p+1$. *See Figure 5.5. Then the inequality, called a* disjoint-partition inequality,

$$x(\delta(V_0, \dots, V_{4p+2})) \geq 4p+3 \tag{5.5}$$

*is valid for* $P_G(D, 2)$.

*Proof.* The following inequalities are valid,

$$x(\delta(V_j)) \geq 2, \; j = 1, \ldots, 4p+2,$$

$$x(\delta(V_0)) \geq 0.$$

By summing these inequalities and dividing by 2, we obtain

$$x(\delta(V_0, V_1, \ldots, V_{4p+2})) \geq 4p+2.$$

Let us suppose that the above inequality holds with equality for an integer solution $\tilde{x}$. Then one should have $\tilde{x}(\delta(V_j)) = 2$ for $j = 1, \ldots, 4p+2$ and $\tilde{x}(\delta(V_0)) = 0$. This implies that $\tilde{x}'$ is the union of disjoint cycles $C_1, \ldots, C_q$ where the nodes correspond to elements of the partition. Here $\tilde{x}'$ is the restriction of $\tilde{x}$ on $G(\bigcup_{i=1,\ldots,4p+2} V_i)$. Note that if a cycle goes through a terminal $s_i$ ($t_i$), then it must also go through $t_i$ ($s_i$). In consequence, each cycle must be even. As $L = 2$ and $|[s_i, t_i]| \leq 1, \forall i = 1, \ldots, 2p+1$, each cycle is of length exactly 4 and therefore uses exactly two pairs of terminals. Thus the cycles altogether go through an even number of pairs of terminals. Since the number of pairs of terminals is odd, this is impossible. □

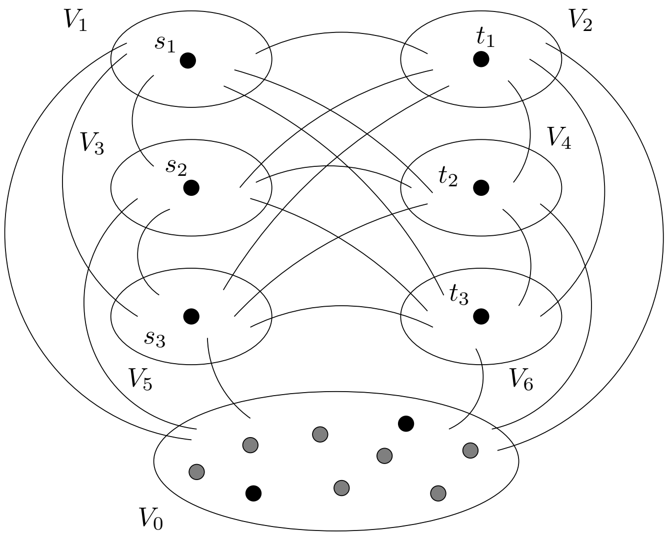


Figure 5.5: Support graph of a disjoint-partition inequality with $p = 1$

## 5.3 New class of valid inequalities for $L = 3$

When $L = 3$, we can extend the family of double path-cut inequalities as follows.

**Theorem 5.3.1.** *Let $V_0, V_1, \ldots, V_5$ be a partition of $N$, $d_1 = \{s_1, t_1\}$ be a demand of $D$ such that $|d_1 \cap V_0| = 1$ and $|d_1 \cap V_5| = 1$, and $V_1$ induce an $s_i t_i$-cut for some $i \in \{2, \ldots, d\}$. Let $e \in [V_1, V_3]$. See Figure 5.6. The* double path-cut inequality*,*

$$x(\delta(V_1)\backslash\{e\}) + x([V_0, V_3 \cup V_4 \cup V_5]) + x([V_2, V_4 \cup V_5]) + x([V_3, V_5]) \geq 3 \tag{5.6}$$

*is valid for $P_G(D, 3)$.*

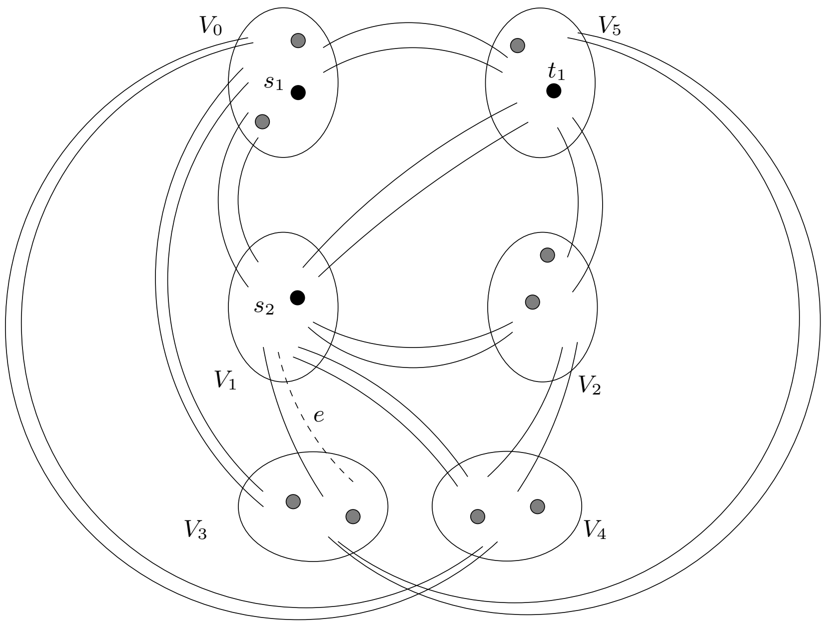


Figure 5.6: Support graph of a double path-cut inequality when $L = 3$ ($t_2 \in N\backslash V_1$)

*Proof.* Inequality (5.6) can be obtained as the rounded-up half-sum of the following constraints:

- $s_i t_i$-cut inequality on subset $V_1$,
- 2-$s_1 t_1$-path-cut inequality on the partition $V_0, V_1 \cup V_2, V_3, V_4, V_5$,
- 2-$s_1 t_1$-path-cut inequality on the partition $V_0, V_2, V_3, V_1 \cup V_4, V_5$,
- trivial inequality $-x(e) \geq -1$,
- trivial inequalities $x(f) \geq 0$ for all $f \in [V_1, V_3]\backslash\{e\}$.

□

## 5.4 Separation procedures

In this section, we present heuristic separation procedures for inequalities (5.1)-(5.6). As before, let $x^*$ be the current solution that we would like to separate.

The separations of inequalities (5.1) and (5.2) are performed simultaneously in the following way. The procedure for separating inequalities (5.6) uses similar ideas. We consider each possible pair of disjoint demands of $D$, say $\{s_1, t_1\}$ and $\{s_2, t_2\}$ with these four nodes distinct two by two. We then look for the edge among $s_1s_2$, $s_1t_2$, $t_1s_2$, $t_1t_2$ with the maximum $x^*$-value. Let $uv$ be this edge and let $u'$ (resp. $v'$) be the terminal associated to $u$ (resp. $v$). All the other nodes besides those four are in $V_3$. We distribute $u, v, u', v'$ to $V_0, V_1, V_2, V_4$ as follows. If the edges between $u$ and $V_3$ have a total weight bigger than those between $v$ and $V_3$, we put $u$ in $V_2$ and $v$ in $V_0$, and hence, $u'$ in $V_1$ and $v'$ in $V_4$. If the contrary holds, we rather put $u$ in $V_0$ and $v$ in $V_2$, while $u'$ and $v'$ are put in $V_4$ and $V_1$, respectively. We then check for the violation of the double path-cut inequality associated to this partition and to edge $e$, where $e$ has the biggest $x^*$-value among the set $[V_1, V_2 \cup V_3]$. Whatever may be, we also try to generate a violated opposed double path-cut inequality. Again, let $uv$ be the edge with the greatest $x^*$-value among $s_1s_2$, $s_1t_2$, $t_1s_2$, $t_1t_2$, and $u', v'$ the terminals associated to $u, v$. This time, if the edges incident to $u'$ and $V_3$ have a bigger weight than those between $v'$ and $V_3$, $u'$ is placed in $V_2$ and $v'$ in $V_0$, and thus $u$ is put in $V_1$ and $v$ in $V_4$. If not, $u'$ is rather placed in $V_0$ and $v'$ in $V_2$, making $u$ in $V_4$ and $v$ in $V_1$. Finally, we test if the opposed double path-cut inequality associated to this partition and to edge $e$, where $e$ has the biggest $x^*$-value among the set $[V_1, V_2 \cup V_3]$, is violated or not by $x^*$.

Inequalities (5.3) and (5.4) are also separated at the same time. This works as follows for any disjoint demands $\{s_1, t_1\}$ and $\{s_2, t_2\}$ in $D$. All the other nodes are first put in $V_3$. For each of these four nodes, we search for the incident edge, to $V_3$ or its associated terminal, with the biggest $x^*$-value. We also compute the total weight of its incident edges to $V_3$ and its associated terminal, minus the previous maximum edge. Let $u$ be the node with this computed value minimum. Then, we put $u$ in $V_1$ and let $e$ be the maximum edge corresponding to $u$. Also, if $u'$ is the terminal associated to $u$, we put $u'$ in $V_2$. The other demand pair is distributed between $V_0$ and $V_4$ according to the total weight of edges between each terminal node and $V_3$ (minimizer in $V_0$, maximizer in $V_4$). We test the violation of the type II inequality corresponding to this partition and edge $e$. In any case, we proceed to the generation of a type I inequality in the following manner. We start from the previous partition and edge $e$, and simply choose the edge $f$ as the maximum edge associated to the terminal in $V_0$. We check if the corresponding type I inequality is violated by $x^*$ or not.

Inequalities (5.5) are separated in the following way. We look for the demand $\{s_1, t_1\}$ of $D$ minimizing the weight of the multicut $V_1 = \{s_1\}$, $V_2 = \{t_1\}$, $V_0 = N \backslash \{s_1, t_1\}$. The corresponding partition is then tested for inducing a violated disjoint-partition inequality. If not, we search for two additional disjoint demands in $V_0$ minimizing the total weight of the edges between two of their four nodes, or between one of them and some node in $V_0$ different from them. The resulting four terminal nodes are taken from $V_0$ to four new susbsets, namely $V_{4p-1}, \ldots, V_{4p+2}$, if this is the $p^{\text{th}}$ iteration of this step. The new disjoint-partition inequality is checked for violation, and, if this is not the case, this procedure is iterated. Of course, we stop whenever there are no longer two disjoint demands remaining in $V_0$.

## 5.5 Computational results

The separation procedures of the previous section were embedded into the Branch-and-Cut algorithm presented in Chapter 4. We would indeed like to test if our new classes of valid inequalities were of any practical use for the THNDP with $L = 2, 3$ and arbitrary demands.

After some preliminary experiments, we noticed that no disjoint-partition inequalities were generated by our separation procedure. Moreover, as already mentioned in the previous chapter, all violated $st$-cut inequalities were found through the heuristic when $L = 2$, that is, no additional ones were produced by the exact separation in that case. Also, inequalities (5.3) and (5.4) were more quickly separated, and more numerously generated, than inequalities (5.1) and (5.2). Finally, for the biggest instances, when demands rarely shared a same node, it seemed more efficient to first separate these new classes of inequalities, than those adapted to the rooted case, that is, the double cut, rooted-(opt-)partition and triple path-cut inequalities from Chapter 4.

As a consequence, we have decided to perform the separation of the following inequalities, and in this precise order,

1. $st$-cut inequalities (heuristically),

2. $L$-$st$-path-cut inequalities,
3. type I and type II inequalities ($L = 2$ only),
4. double path-cut inequalities ($L = 2, 3$) and opposed double path-cut inequalities ($L = 2$ only),
5. triple path-cut inequalities,
6. double cut inequalities,
7. rooted- (opt-, when $L = 2$) partition inequalities,
8. $st$-cut inequalities (exactly, when $L = 3$ only).

In the first two tables here below, we present average results obtained for the same random instances with $L = 2, 3$ and arbitrary demands as in Section 4.7. Here we only separate the basic and "disjoint" inequalities in the order 1. to 4., eventually plus 8. if $L = 3$.

The entries are as follows:

| | | |
|---|---|---|
| $n$ | : | the number of nodes of the problem, |
| $d$ | : | the number of demands, |
| Cu | : | the number of generated $st$-cut inequalities, |
| Pc | : | the number of generated $L$-path-cut inequalities, |
| Dp | : | the number of generated double path-cut inequalities, |
| Dpo | : | the number of generated opposed double path-cut inequalities, |
| TI | : | the number of generated type I inequalities, |
| TII | : | the number of generated type II inequalities, |
| o3 | : | the number of problems solved to optimality over the five instances tested, |
| Gap3 | : | the gap between the best upper bound and the lower bound obtained at the root node of the Branch-and-Cut tree, |
| Gt3 | : | the gap between the best upper bound and the best lower bound, |
| Tree3 | : | the number of nodes in the Branch-and-Cut tree, |
| CPU3 | : | the total time in seconds. |

Let us compare Table 4.3 from Chapter 4 with Table 5.1. First, one can see that the same number of instances were solved to optimality within the time limit (5 hours). For these ones, the gap at the root node (Gap3) is similar to the one obtained with

Table 5.1: Results for random instances for $L = 2$ and arbitrary demands, with basic and disjoint inequalities only

| $n$ | $d$ | Cu | Pc | Dp | Dpo | TI | TII | o3 | Gap3 | Gt3 | Tree3 | CPU3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 5 | 4 | 60 | 21 | 16 | 40 | 51 | 5 | 8.9 | 0.0 | 22 | 0.28 |
| 20 | 5 | 2 | 77 | 11 | 7 | 49 | 48 | 5 | 3.8 | 0.0 | 15 | 0.49 |
| 20 | 10 | 7 | 816 | 71 | 43 | 99 | 177 | 5 | 8.0 | 0.0 | 558 | 66.82 |
| 30 | 8 | 6 | 626 | 67 | 58 | 192 | 244 | 5 | 8.5 | 0.0 | 425 | 45.16 |
| 30 | 15 | 12 | 5319 | 390 | 420 | 341 | 1080 | 2 | 11.8 | 3.5 | 15063 | 11489.65 |
| 40 | 10 | 6 | 531 | 47 | 36 | 134 | 178 | 5 | 5.1 | 0.0 | 109 | 22.25 |
| 40 | 20 | 13 | 6473 | 303 | 282 | 261 | 901 | 0 | 19.9 | 14.9 | 8211 | 18000.00 |

the basic and "rooted" inequalities (Gap2), and thus, quite less than the one obtained with the basic constraints only (Gap1). In the same way, more CPU time is used when more inequalities are separated, which yields a total time CPU3 bigger than CPU1 (and similar to CPU2). For instances not solved to optimality, one can remark contradicting results. For $(n, d) = (30, 15)$, we have indeed that the gap at the root node (Gap3) and the final gap (Gt3) are both reduced by about half a percent compared to the previous ones (Gap1, Gap2, Gt1, Gt2). Also, the average CPU time has decreased, which makes the new classes of disjoint inequalities quite better than the rooted inequalities introduced in Chapter 4. However, when $(n, d) = (40, 20)$, all these values become worse than their counterparts obtained in the previous tests. This is mainly due to the fact that, if the lower bounds slightly improve in this new run, the upper bounds are not as good as before (since more time is now passed in separations and less in branchings, see Tree3 compared to Tree1 and Tree2), making the overall gaps larger.

Table 5.2: Results for random instances for $L = 3$ and arbitrary demands, with basic and disjoint inequalities only

| $n$ | $d$ | Cu | Pc | Dp | o3 | Gap3 | Gt3 | Tree3 | CPU3 |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 5 | 9 | 88 | 52 | 5 | 7.6 | 0.0 | 37 | 0.49 |
| 20 | 5 | 15 | 703 | 159 | 5 | 8.4 | 0.0 | 111 | 15.97 |
| 20 | 10 | 27 | 21568 | 2552 | 2 | 19.5 | 6.6 | 4521 | 11983.82 |
| 30 | 8 | 39 | 14694 | 2183 | 3 | 15.3 | 3.0 | 2020 | 8712.69 |
| 30 | 15 | 26 | 20258 | 1450 | 0 | 56.6 | 53.6 | 1092 | 18000.00 |
| 40 | 10 | 34 | 16533 | 1722 | 0 | 32.8 | 27.2 | 1516 | 18000.00 |
| 40 | 20 | 23 | 12333 | 494 | 0 | 61.5 | 60.2 | 269 | 18000.00 |

For $L = 3$, the double path-cut inequalities introduced in this chapter do not give very satisfactory results. Indeed, the gaps and CPU time are between the corresponding values obtained in Chapter 4 thanks to the basic constraints, respectively with and without the rooted inequalities. Only for $(n, d) = (20, 10)$, the final gap here (Gt3) is slightly smaller than both of the previous ones (Gt1,Gt2).

The following tables present average results on the same set of THPP instances, but, this time, the "rooted" inequalities are also separated (in the order 1. to 8.). We have the following entries.

| | | |
|---|---|---|
| $n$ | : | the number of nodes of the problem, |
| $d$ | : | the number of demands, |
| Cu | : | the number of generated $st$-cut inequalities, |
| Pc | : | the number of generated 2-path-cut inequalities, |
| Dc | : | the number of generated double cut inequalities, |
| Ro | : | the number of generated rooted-(opt-)partition inequalities, |
| Tp | : | the number of generated triple path-cut inequalities, |
| Dp | : | the number of generated double path-cut inequalities, |
| Dpo | : | the number of generated opposed double path-cut inequalities, |
| TI | : | the number of generated type I inequalities, |
| TII | : | the number of generated type II inequalities, |
| o4 | : | the number of problems solved to optimality over the five instances tested, |
| Gap4 | : | the gap between the best upper bound and the lower bound obtained at the root node of the Branch-and-Cut tree, |
| Gt4 | : | the gap between the best upper bound and the best lower bound, |
| Tree4 | : | the number of nodes in the Branch-and-Cut tree, |
| CPU4 | : | the total time in seconds. |

Table 5.3: Results for random instances for $L = 2$ and arbitrary demands, with basic, disjoint and rooted inequalities

| $n$ | $d$ | Cu | Pc | Dc | Ro | Tp | Dp | Dpo | TI | TII | o4 | Gap4 | Gt4 | Tree4 | CPU4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 5 | 3 | 57 | 5 | 3 | 9 | 16 | 14 | 39 | 50 | 5 | 8.7 | 0.0 | 22 | 0.28 |
| 20 | 5 | 1 | 69 | 3 | 3 | 5 | 12 | 7 | 44 | 47 | 5 | 3.8 | 0.0 | 11 | 0.47 |
| 20 | 10 | 7 | 833 | 27 | 20 | 92 | 60 | 44 | 81 | 151 | 5 | 7.8 | 0.0 | 521 | 73.74 |
| 30 | 8 | 6 | 594 | 22 | 20 | 109 | 58 | 49 | 181 | 218 | 5 | 8.5 | 0.0 | 289 | 34.34 |
| 30 | 15 | 12 | 5299 | 203 | 112 | 901 | 330 | 363 | 318 | 980 | 2 | 11.6 | 3.5 | 14001 | 11719.93 |
| 40 | 10 | 7 | 521 | 14 | 8 | 50 | 38 | 32 | 130 | 166 | 5 | 5.0 | 0.0 | 80 | 22.08 |
| 40 | 20 | 13 | 6341 | 151 | 16 | 606 | 247 | 287 | 240 | 859 | 0 | 21.9 | 17.1 | 7929 | 18000.00 |

Let us now analyze Table 5.3 compared to the previous ones. One can first remark that a large amount of violated inequalities were generated in all families (basic, disjoint and rooted). This means that, in the case of arbitrary demands, the interaction between two intersecting demands is still to consider. Of course, more disjoint inequalities are found than rooted ones, but the difference is not so important. Compared to the results presented in Chapter 4, one can draw similar conclusions as for Table 5.1. This new run still slightly improves the gaps for all instances up to $(n, d) = (40, 10)$. The contrary unfortunately happens for the bigger instances. The additional time used in separations limits again the search in the Branch-and-Cut tree. This has an impact on the quality of the final upper bounds, which annihilates the improvement made on the lower bounds.

Table 5.4: Results for random instances for $L = 3$ and arbitrary demands, with basic, disjoint and rooted inequalities

| $n$ | $d$ | Cu | Pc | Dc | Ro | Tp | Dp | o4 | Gap4 | Gt4 | Tree4 | CPU4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 5 | 7 | 67 | 21 | 3 | 2 | 37 | 5 | 5.3 | 0.0 | 26 | 0.43 |
| 20 | 5 | 20 | 556 | 46 | 5 | 0 | 136 | 5 | 7.6 | 0.0 | 89 | 11.65 |
| 20 | 10 | 30 | 20311 | 385 | 4 | 4 | 1811 | 2 | 19.2 | 7.4 | 4395 | 11407.33 |
| 30 | 8 | 38 | 14147 | 272 | 22 | 3 | 1278 | 3 | 13.6 | 1.7 | 1896 | 8432.54 |
| 30 | 15 | 28 | 19650 | 297 | 0 | 0 | 822 | 0 | 56.4 | 53.4 | 1196 | 18000.00 |
| 40 | 10 | 35 | 16491 | 272 | 1 | 1 | 1138 | 0 | 29.3 | 23.4 | 1634 | 18000.00 |
| 40 | 20 | 28 | 12213 | 132 | 0 | 1 | 312 | 0 | 61.4 | 60.0 | 289 | 18000.00 |

For $L = 3$, the simultaneous use of basic, rooted and disjoint inequalities is quite effective compared to their separate use. We indeed get an improvement of the gap at the root node and of the final gap in nearly all cases. For example, when $(n, d) = (30, 8)$, Gt4 is equal to 1.7%, which is 1.4% smaller than Gt1 and 0.4% smaller than Gt2 (see Table 4.4). Also, when $(n, d) = (40, 10)$, Gt4 has for value 23.4%, that is, 0.4% less than Gt1 and 4.8% less than Gt2. Finally, note that these good results are mainly due to the basic constraints, the double cut inequalities and the double path-cut inequalities, since very few violated constraints were found among the other classes of rooted inequalities, namely the triple path-cut and rooted-partition ones.

## 5.6 Concluding remarks

In this chapter, we have proposed several additional classes of valid inequalites for the THNDP with $L = 2, 3$. These are based on disjoint demands, that is, demands that do no share nodes. We have then given heuristic procedures to separate these inequalities in order to use them within the framework of the Branch-and-Cut algorithm devised in Chapter 4.

From the new computational results obtained, it appears that separating these inequalities and the previous "rooted" ones at the same time yields the best improvements concerning the gaps. This is especially true when $L = 3$, since, in that case, the double path-cut inequalities do not give satisfactory results when used alone.

Of course, additional improvements could still be performed. First, one could study necessary and sufficient conditions for inequalities (5.1)-(5.6) to be facet-defining. This would indeed permit to devise more effective separation procedures, in the sense that they would only generate the facially best cuts. On the other hand, we think that some work could still be done concerning the obtention of better upper bounds. For the biggest instances, random rounding indeed begins to reach its limits. It would be interesting to implement further heuristics for obtaining feasible solutions.

# Chapter 6

# Two Disjoint 4-Hop-Constrained Paths Problems

In this chapter, we pursue the study of the THPP started in Chapter 2. This time, we also consider a variant of this problem, namely where the two required $L$-$st$-paths of the solution must be node-disjoint instead of edge-disjoint. For both versions, we give an integer programming formulation in the space of the design variables when $L = 4$. This work has been the object of an article with A. Ridha Mahjoub [46], although not yet published.

## 6.1 Introduction

Recall the following definition from Chapter 2. Given a function $c : E \to I\!R$ which associates a cost $c(e)$ to each edge $e \in E$, the *Two Hop-constrained Paths Problem* (THPP) is to find a minimum cost subgraph such that between $s$ and $t$ there exist at least two $L$-$st$-paths. In this chapter, we will speak of the *node* THPP if those two paths must be node-disjoint, and of the *edge* THPP if they must be edge-disjoint.

In what follows, we consider the node THPP and the edge THPP when $L = 4$. For both versions, we give an integer programming formulation in the space of the design variables. Such a formulation is indeed not trivial to obtain for the edge case when $L = 4$, as already mentioned in Chapter 2.

In the next section, we first propose an integer programming formulation, in the space of the design variables, for the node THPP with $L = 4$. This one will also be valid when $L = 2, 3$. It will be based not only on the $st$-cut and $L$-$st$-path-cut inequalities introduced before, but also on "node-versions" of these ones. In Section 6.3, we first recall why the formulation known for the edge THPP when $L = 2, 3$ is no more valid when $L = 4$. We then present a new family of valid inequalities for the edge THPP when $L = 4$. Using this family, we give an integer programming formulation, in the space of the design variables, in that case. Finally, in Section 6.4, we present our concluding remarks.

## 6.2 Formulation for the node THPP when $L = 4$

From Chapter 2, we know that the incidence vector $x^F$ of any solution $(N, F)$ to the THPP satisfies the following inequalities.

$$\begin{aligned} &x(\delta(W)) \geq 2, && \text{for all } st\text{-cut } \delta(W), \\ &x(T) \geq 2, && \text{for all } L\text{-}st\text{-path-cut } T, \\ &1 \geq x(e) \geq 0, && \text{for all } e \in E. \end{aligned}$$

These inequalities are called *st-cut inequalities*, *L-(st-)path-cut inequalities*, and *trivial inequalities*, respectively.

Moreover, in the node case, the following inequalities, which we call *st-node-cut inequalities*, are also valid for the problem. They are defined as

$$x_{G-z}(\delta(W)) \geq 1,$$

where $z \in N \backslash \{s, t\}$ and $\delta(W)$ is an $st$-cut in the graph $G - z$.

In the same way, we can write *L-path-node-cut inequalities*, that is,

$$x_{G-z}(T) \geq 1,$$

where $z \in N \backslash \{s, t\}$ and $T$ is an $L$-path-cut in the graph $G - z$, which are valid for the node THPP.

Finally, consider the following linear system consisting of the inequalities introduced above, that is,

$$
\begin{aligned}
& x(\delta(W)) \geq 2, && \text{for all } st\text{-cut } \delta(W), && (6.1)\\
& x_{G-z}(\delta(W)) \geq 1, && \text{for all } st\text{-cut } \delta(W), \text{ for all } z \in N\backslash\{s,t\}, && (6.2)\\
& x(T) \geq 2, && \text{for all } L\text{-path-cut } T, && (6.3)\\
& x_{G-z}(T) \geq 1, && \text{for all } L\text{-path-cut } T, \text{ for all } z \in N\backslash\{s,t\}, && (6.4)\\
& x(e) \leq 1, && \text{for all } e \in E, && (6.5)\\
& x(e) \geq 0, && \text{for all } e \in E. && (6.6)
\end{aligned}
$$

We will show that the system (6.1)-(6.6), along with the integrality constraints, formulates the node THPP as an integer program when $L = 4$.

**Theorem 6.2.1.** *The node THPP for* $L = 4$ *is equivalent to the integer program*

$$\textit{Min } \{cx \,:\, x \textit{ satisfies (6.1)-(6.6)}, x \in \mathbb{Z}^E\}.$$

*Proof.* **Necessity.** First we show that any feasible solution $(N, F)$ to the node THPP with $L = 4$ has an incidence vector $x^F$ satisfying constraints (6.1)-(6.6).
Let $G_F$ be the support graph of $x^F$. First, it is clear that the constraints $0 \leq x(e) \leq 1$ are all satisfied by definition of the boolean vector $x^F$. Suppose that there exists a subset of nodes $W$, containing $s$ and not $t$, such that $x^F(\delta(W)) \leq 1$. Then there is at most one edge $e \in \delta(W)$ such that $x^F(e) = 1$, and all the $st$-paths in $G_F$ must go through $e$. This contradicts the existence of two node-disjoint $st$-paths. Similarly, if $x^F_{G-z}(\delta(W)) = 0$, it means that there is no $st$-path in $G_F - z$, and therefore, that all the $st$-paths in $G_F$ go through $z$, a contradiction. Now, if there exists a 4-path-cut $T$ such that $x^F(T) \leq 1$, there is at most one edge $e \in T$ with $x^F(e) = 1$. If one $st$-path of $G_F$ uses $e$, then the other can only go through edges of $F\backslash T$. Clearly, the minimum number of hops of such a path is 5, contradicting the feasibility of the solution. Finally, if $x^F_{G-z}(T) = 0$, the shortest $st$-path in $G_F - z$ is of length at least 5. Hence, in $G_F$, all the 4-$st$-paths go through $z$, which is impossible.

**Sufficiency.** Consider an edge subset $F \subseteq E$ and suppose that $F$ does not induce a solution to the node THPP with $L = 4$. Suppose that all $st$-cut and $st$-node-cut constraints are satisfied by $x^F$. We are going to show that there is at least one 4-path-cut or 4-path-node-cut violated by $x^F$. Let $G_F$ be the graph induced by $F$ and $P_0$ a shortest $st$-path (in number of hops) in $G_F$. In what follows, we are going to discuss different cases with respect to the length of $P_0$.

If $|P_0| = 1$, that is $P_0 = (st)$ with $st \in [s,t]$, then $P_0$ is the only 4-$st$-path in $G_F$. In fact, if there exists a 4-$st$-path $P$ different from $st$, then $P$ would be node-disjoint from $st$, a contradiction. Therefore, in $G_F - st$, since the $st$-cut inequalities are satisfied, there must exist an $st$-path, of length at least 5. Let us define $V_i$, $i = 0, \dots, 4$, as the subset of nodes at distance $i$ from $s$ in $G_F - st$, and $V_5 = N \backslash (\bigcup_{i=0}^{4} V_i)$. By the previous remarks, it is clear that the $V_i$'s are nonempty, and that $s \in V_0$ and $t \in V_5$. Moreover, by construction, no edge of $G_F - st$ is in the corresponding 4-path-cut $T$, and hence, $x^F_{G-st}(T) = 0$. Therefore, in $G$, we obtain that $x^F(T) = 1$, which is a violated 4-path-cut inequality.

If $|P_0| = 2$, that is $P_0 = (su, ut)$ with $u \in N \backslash \{s,t\}$, since $F$ is not a solution to the problem, all the other 4-$st$-paths of $G_F$ must go through $u$. Therefore, in $G_F - u$, since the $st$-node-cut inequalities are satisfied, there exists an $st$-path, of length at least 5. Let us define a 4-path-cut $T$ in $G_F - u$ along the same way as in the previous case. By construction, no edge of $G_F - u$ is in $T$, and hence, $x^F_{G-u}(T) = 0$. This implies that the 4-path-node-cut inequality corresponding to $T$ and $u$ is violated.

If $|P_0| = 3$, that is $P_0 = (su, uv, vt)$ with $u, v \in N \backslash \{s,t\}$, $u \neq v$, then all the other 4-$st$-paths of $G_F$ must go through either $u$, or $v$, or both. Suppose that there exist at least one 4-$st$-path $P_1$ going through $u$, but not $v$, and at least one 4-$st$-path $P_2$ going through $v$, but not $u$. Let $P_1^t$ (resp. $P_1^s$) be the subpath of $P_1$ between $u$ and $t$ (resp. $s$), and $P_2^s$ (resp. $P_2^t$) the subpath of $P_2$ between $s$ (resp. $t$) and $v$. Clearly, $|P_1^t| \leq 3$ and $|P_2^s| \leq 3$. Moreover, since $P_0$ is a shortest $st$-path, we have that $|P_1^t| \geq 2$ and $|P_2^s| \geq 2$. Hence, $1 \leq |P_1^s| \leq 2$ and $1 \leq |P_2^t| \leq 2$. Now, since $F$ is not feasible, $P_1$ and $P_2$ must intersect each other in a node $w$ different from $s, u, v, t$. If $w \in P_1^t \cap P_2^s$, then $P_0$ and the 4-$st$-path consisting of the subpath of $P_2$ from $s$ to $w$, and of the subpath of $P_1$ from $w$ to $t$, are node-disjoint, a contradiction. Therefore, either $w \in P_1^s$ or $w \in P_2^t$. Suppose that $w \in P_1^s$ (the other case follows by symmetry). Then $|P_1^s| = 2$ and hence $|P_1^t| = 2$. Let $P_2^{vw}$ be the subpath of $P_2$ between $v$ and $w$. Clearly, $|P_2^{vw}| \leq 2$. But then, we have two node-disjoint 4-$st$-paths in $G_F$, namely $su, P_1^t$, and $sw, P_2^{vw}, vt$. And this contradicts the infeasibility of the solution. Consequently, there cannot be at the same time 4-$st$-paths going only through $u$, or through $v$. Suppose w.l.o.g. that $u$ belongs to all 4-$st$-paths of $G_F$ (the other case is symmetric). Therefore, there exists no 4-$st$-path in $G_F - u$ and, by constructing the 4-path-node-cut $T$ as before, we get a contradiction.

Now suppose that $|P_0| = 4$, that is $P_0 = (s, u, v, w, t)$ with $u, v, w \in N\backslash\{s, t\}$, $u, v, w$ different from each other. Then all the other 4-$st$-paths of $G_F$ must go through either $u$, or $v$, or $w$.

*Claim.* There does not exist a 4-$st$-path going through $u$ which does not use neither $v$ nor $w$.

*Proof.* Assume the contrary, and let $P$ be a 4-$st$-path that contains $u$, but neither $v$, nor $w$. Since $P_0$ is a shortest path, we have that $P$ must contain three edges between $u$ and $t$. Let $v'$, $w'$ be the nodes along this subpath, different from $u$ and $t$. Now, suppose that there exists a 4-$st$-path $P'$ not going through $u$, but through $v$ or $w$. Thus, $P'$ must also intersect $P$ in either $v'$ or $w'$. Since $P_0$ is a shortest $st$-path, it is not hard to see that we have either $P' = (s, u', v', w, t)$ or $P' = (s, u', v, w', t)$, with $u' \in N\backslash\{s, u, v, w, v', w', t\}$. But then, the graph induced by $P_0 \cup P \cup P'$ contains two node-disjoint 4-$st$-paths, a contradiction. Consequently, there cannot exist a 4-$st$-path not going through $u$, and hence, all 4-$st$-paths of $G_F$ use $u$. This implies that $G_F - u$ does not contain 4-$st$-paths. We can then get a contradiction in a similar way as before. ♦

By the claim above, all 4-$st$-paths in $G_F$ go through $v$ or $w$. Suppose there is a 4-$st$-path $\tilde{P}$ going through $w$, but not $v$. If $\tilde{P}$ does not contain $u$, we can show along the same line that all 4-$st$-paths use $w$, and get a violated 4-path-node-cut inequality.

So suppose that $\tilde{P}$ contains $u$. We may also suppose that there are two further 4-$st$-paths $P_1$ and $P_2$ such that $P_1$ (resp. $P_2$) uses $u$ but not $w$ (resp. $w$ but not $u$). For otherwise, we would have that either each 4-$st$-path contains $u$ or each 4-$st$-path contains $w$. In both cases, we would get as before a violated 4-path-node-cut. Moreover, $P_1$ and $P_2$ must go through $v$, for otherwise, we would get two node-disjoint 4-$st$-paths, a contradiction. Hence, $P_1$ is of the form $(s, u, v, P_1^t)$ where $P_1^t$ is a $vt$-path of length 2 not going through $w$, and $P_2$ is of the form $(P_2^s, v, w, t)$ where $P_2^s$ is a $sv$-path of length 2 not going through $u$. But then, we obtain two node-disjoint 4-$st$-paths, namely $P_2^s \cup P_1^t$ and $\tilde{P}$, which is a contradiction.

In consequence, all 4-$st$-paths go through $v$. As before, we obtain a violated 4-path-node-cut inequality.

If $|P_0| \geq 5$, there exists no 4-$st$-path in $G_F$. Thus, we can build directly an adequate partition in $G_F$ and get a violated 4-path-cut inequality. □

If this result clearly holds for $L = 2, 3$ by doing a similar proof, this is not the case for $L = 5$. Consider indeed the integer solution in Figure 6.1. We have that its incidence vector satisfies inequalities (6.1)-(6.6). However, this solution is clearly infeasible for the node THPP with $L = 5$.

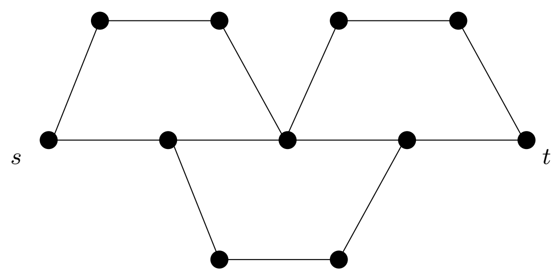


Figure 6.1: An infeasible solution to the node THPP for $L = 5$, but satisfying system (6.1)-(6.6)

## 6.3 Formulation for the edge THPP when $L = 4$

In Chapter 2, we have shown that the linear system of inequalities (6.1),(6.3),(6.5),(6.6), along with the integrality constraints, is sufficient to formulate the edge THPP for $L = 2, 3$. However, this is not the case when $L = 4$, as illustrated by Figure 6.2. One can verify that all those inequalities are satisfied, while the solution is not feasible for the edge THPP with $L = 4$.

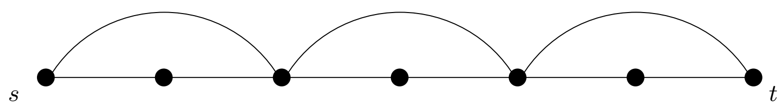


Figure 6.2: An infeasible solution to the edge THPP for $L = 4$, but satisfying inequalities (6.1),(6.3),(6.5),(6.6)

In [16], Dahl and Gouveia describe the following class of valid inequalities for the $L$-hop-constrained path problem. Let $V_0, \ldots, V_{L+r}$ be a partition of $N$ such that $r \geq 1$, $s \in V_0$ and $t \in V_{L+r}$. Then, the *generalized jump inequality* is

$$\sum_{e \in [V_i, V_j], i \neq j} \min(|i-j|-1, r)\, x(e) \geq r.$$

These inequalities can be easily extended to the edge THPP, as follows,

$$\sum_{e \in [V_i, V_j], i \neq j} \min(|i-j|-1, r)\, x(e) \geq 2r.$$

Note that these inequalities generalize the $L$-path-cut inequalities (by setting $r = 1$). Moreover, they permit to eliminate the previous counterexample, namely by taking $r = 2$ and each $V_i$ singleton. However, adding this class is still not sufficient to formulate the edge THPP for $L = 4$. See for example Figure 6.3. Clearly, this solution is not feasible for the edge THPP with $L = 4$. However, it is not difficult to check that, besides all $st$-cut and trivial constraints, its incidence vector also satisfies all generalized jump inequalities.

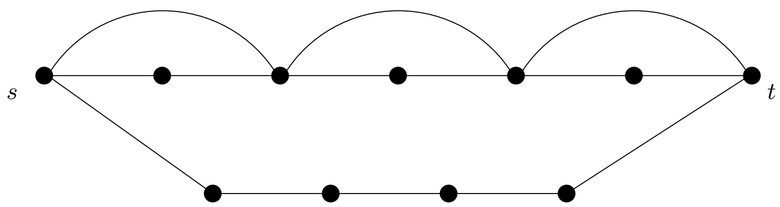


Figure 6.3: An infeasible solution to the edge THPP for $L = 4$, but satisfying the previous system plus the generalized jump inequalities

So, in order to formulate the edge THPP for $L = 4$, we propose the following new class of inequalities, which we call *two-layered 4-path-cut inequalities*. Let $V_0, V_1, \ldots, V_6$, $W_1, \ldots, W_4$ be a two-layered partition of $N$ such that $s \in V_0$ and $t \in V_6$. See Figure 6.4 for an illustration. We have that all the $V_i$'s are nonempty, and that the $W_i$'s may be empty or not. The associated inequality is denoted by

$$ax \geq 4, \tag{6.7}$$

with

$$
\begin{aligned}
&a(e) = \min(|i-j|-1, 2), && \forall e \in [V_i, V_j], i \neq j,\\
&a(e) = 2, && \forall e \in [W_i, W_j], |i-j| \geq 2,\\
&a(e) = 2, && \forall e \in [V_i, W_j], j-i \geq 2 \text{ or } i-j \geq 3,\\
&a(e) = 1, && \forall e \in [V_i, W_j], (i,j) = (2,3), (3,1), (3,4), (4,2),\\
&a(e) = 0, && \text{if not.}
\end{aligned}
$$

In Figure 6.4, the edges with coefficient 1 are in solid lines, while those with coefficient 2 are in bold. The edges with zero coefficient do not appear in the figure.

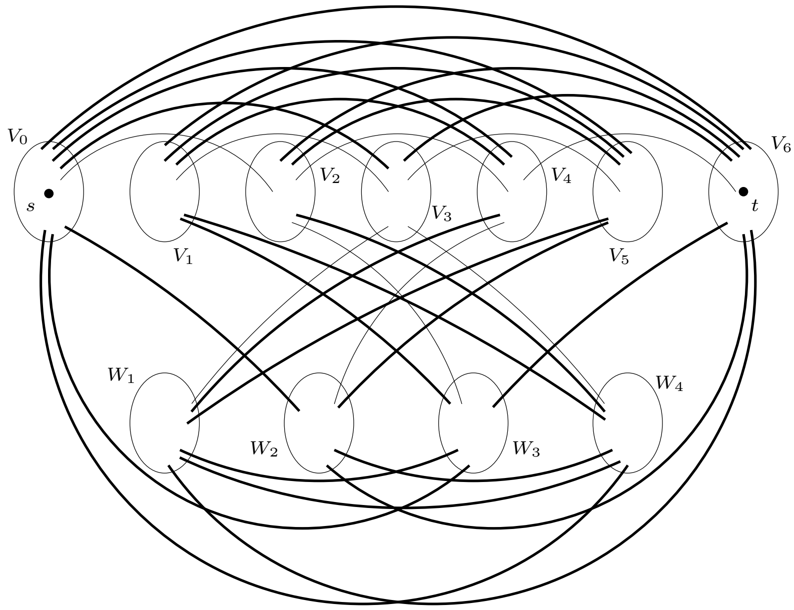


Figure 6.4: Support graph of a two-layered 4-path-cut inequality

**Theorem 6.3.1.** *Inequalities (6.7) are valid for the edge THPP polytope when $L = 4$.*

*Proof.* Consider a two-layered 4-path-cut inequality $ax \geq 4$, whose coefficients can be seen as weights on the edges. It is easily seen that the total weight of any 4-$st$-path is at least 2. Since any feasible solution to the edge THPP with $L = 4$ must contain at least two such edge-disjoint paths, we have that its incidence vector satisfies $ax \geq 4$. □

Observe that, when the $W_i$'s are all empty, these inequalities correspond to the generalized jump inequalities with $r = 2$. But now, the counterexample presented in Figure

6.3 can be cut off by the two-layered 4-path-cut inequality obtained from the partition with all the sets being single nodes.

We are now going to show that the linear system consisting of inequalities (6.1),(6.3),(6.5),(6.6),(6.7), along with the integrality constraints, is sufficient to formulate the edge THPP with $L = 4$.

**Theorem 6.3.2.** *The edge THPP for* $L = 4$ *is equivalent to the integer program*

$$\text{Min } \{cx : x \text{ satisfies (6.1),(6.3),(6.5),(6.6),(6.7)}, x \in \mathbb{Z}^E\}.$$

*Proof.* **Necessity.** By Theorem 6.3.1 together with the fact that the $st$-cut, $L$-path-cut, and trivial constraints are valid for the edge THPP polytope, we have that the incidence vector of any solution to the edge THPP for $L = 4$ satisfies inequalities (6.1),(6.3),(6.5),(6.6),(6.7).

**Sufficiency.** Suppose there exists a solution $F$ whose incidence vector $x^F$ satisfies the $st$-cut, 4-path-cut and trivial inequalities, but that is not feasible for the edge THPP with $L = 4$. We will show that there exists a two-layered 4-path-cut inequality $ax \geq 4$ violated by $x^F$. Let $G_F$ be the graph induced by $F$ and $P_0$ a shortest $st$-path in $G_F$. We have the following claims.

*Claim 1.* For every edge $e$, there is at least one 4-$st$-path in $G_F - e$.

*Proof.* Suppose this is not the case. Let $e_0$ be an edge such that $G_F - e_0$ does not contain any 4-$st$-path. Since $x^F$ satisfies all the $st$-cut inequalities, there must still exist an $st$-path in $G_F - e_0$, of length at least 5. Let us consider a partition $V_0, \dots, V_5$ of $N$ such that $V_i$, $i = 0, \dots, 4$, contains all the nodes at distance $i$ from $s$ in $G_F - e_0$, and $V_5$ contains all the other nodes. By the previous remark, we have that each $V_i$ is nonempty, $s \in V_0$, and $t \in V_5$. Let $T$ be the associated 4-path-cut. By construction, $x^F_{G-e_0}(T) = 0$, and hence, $x^F(T) \leq 1$, a contradiction. ♦

*Claim 2.* The path $P_0$ is of length at least 3.

*Proof.* Suppose this is not the case, that is, either $|P_0| = 1$, or $|P_0| = 2$.
If $|P_0| = 1$, then $P_0 = (st)$. By Claim 1, there is a 4-$st$-path that does not contain $st$.

But this implies that $G_F$ contains two edge-disjoint 4-$st$-paths, a contradiction.
Thus $|P_0| = 2$. Let $P_0 = (e_1, e_2)$. By Claim 1, there is a 4-$st$-path $P_1$ that does not contain $e_1$. Then $e_2 \in P_1$. Let $P_1^s$ be the subpath of $P_1$ between $s$ and $v$, where $v$ is the end node of $e_1$ different from $s$. Hence $|P_1^s| \leq 3$. Similarly, there is a 4-$st$-path $P_2$ such that $e_1 \in P_2$ and $e_2 \notin P_2$. Let $P_2^t$ be the subpath between $v$ and $t$. We also have that $|P_2^t| \leq 3$.
If $P_1^s$ and $P_2^t$ do not intersect in some edge, then the paths $e_1, P_2^t$ and $P_1^s, e_2$ are of length at most 4 and edge-disjoint, a contradiction.
Let $w$ be the first node common to $P_1^s$ and $P_2^t$, $w \neq v$. Let $\tilde{P}_2^t$ be the subpath of $P_2^t$ between $w$ and $t$. Clearly, $|\tilde{P}_2^t| \leq 2$. Similarly, let $\tilde{P}_1^s$ be the subpath of $P_1^s$ between $s$ and $w$. We also have $|\tilde{P}_1^s| \leq 2$. Now let $P_0'$ be the path $\tilde{P}_1^s \cup \tilde{P}_2^t$. We have that $|P_0'| \leq 4$ and $P_0 \cap P_0' = \phi$, a contradiction. ♦

*Claim 3.* The path $P_0$ is of length exactly 3.

*Proof.* By Claim 2, we already have that $|P_0| \geq 3$. Also, it is clear that $|P_0| \leq 4$. If not, this would contradict Claim 1 for any edge $e$. Suppose now, by contradiction, that $|P_0| = 4$.
Let $P_0 = (sv_1, v_1v_2, v_2v_3, v_3t)$ with $v_1, v_2, v_3 \in N\backslash\{s, t\}$, $v_1, v_2, v_3$ different from each other. By Claim 1, there must exist a 4-$st$-path $P_1$ not containing $sv_1$, and another one, $P_2$, not containing $v_3t$. Since $P_0$ is a shortest path, $P_1$ and $P_2$ are both of length exactly 4. Let $P_1^s$ (resp. $P_2^t$) be the subpath of $P_1$ (resp. $P_2$) from $s$ (resp. $t$) to the first node common with $P_0$. Note that, if this node is $v_i$, then $|P_1^s| = i$ (resp. $|P_2^t| = 4 - i$). Moreover, note that $P_1^s$ and $P_2^t$ do not intersect $P_0$ in any edge.
Suppose first that $P_1^s$ and $P_2^t$ intersect in some edge. Thus there is a node $z$ common to $P_1^s$ and $P_2^t$, different from $v_1, v_2, v_3$. But then the subpaths of $P_1^s$, $P_2^t$ between $z$ and $s$, $t$, respectively, form a 4-$st$-path disjoint from $P_0$, a contradiction. Therefore, $P_1^s$ and $P_2^t$ are edge-disjoint.
Now remark that $P_1^s$ cannot go from $s$ to $v_3$. If this was the case, we would indeed have $|P_1^s| = 3$, and hence, $P_1$ would be of the form $sv_1', v_1'v_2', v_2'v_3, v_3t$. But then, as $v_3t \notin P_2$, and $P_1^s$ and $P_2^t$ are edge-disjoint, $P_2\backslash P_2^t$ must have an edge in common with $P_1^s$ (and also with $P_0$). Clearly, the only possibility is that $P_2$ is of the form $(sv_1', v_1'v_2, v_2v_3, h)$ or $(sv_1, v_1v_2', v_2'v_3, h)$, with $h \in [v_3, t]\backslash\{v_3t\}$. But this creates two edge-disjoint 4-$st$-paths, a contradiction. Similarly, we have that $P_2^t$ cannot go from $t$ to $v_1$.
Suppose now that $P_1^s$ goes from $s$ to $v_2$. Therefore, in order to not create two edge-disjoint 4-$st$-paths, $P_2^t$ must go from $t$ to $v_3$. Recall that, in this case, $P_1^s = (sv_1', v_1'v_2)$ and $P_2^t = (h)$ with $h \in [v_3, t]\backslash\{v_3t\}$. Consequently, it is clear that $v_2v_3$ must be common to $P_0, P_1, P_2$. On the other hand, by Claim 1, there must exist a further 4-$st$-path

not containing $v_2v_3$. However, since this one cannot intersect all the previous paths in at least one edge, we obtain a contradiction.
Thus, $P_1^s$ must go from $s$ to $v_1$ and, by symmetry, $P_2^t$ must go from $t$ to $v_3$. Hence, $P_1^s = (g)$ with $g \in [s, v_1]\backslash\{sv_1\}$, and $P_2^t = (h)$ with $h \in [v_3, t]\backslash\{v_3t\}$. Note that a 4-$st$-path cannot intersect at the same time $sv_1$ and $g$, or $v_3t$ and $h$. Therefore, if we consider a 4-$st$-path not containing $v_1v_2$, it must use $v_2v_3$. In the same way, a 4-$st$-path not containing $v_2v_3$ must use $v_1v_2$. Once again, we obtain two edge-disjoint 4-$st$-paths.♦

In the rest of the proof, we let $P_0 = (e_1, e_2, e_3) = (s, v_1, v_2, t)$.

*Claim 4.* Every 4-$st$-path of $G_F$ contains at least two edges among $e_1, e_2, e_3$.

*Proof.* Let $P$ be a 4-$st$-path different from $P_0$. Let us suppose that $P$ does not intersect $\{e_1, e_2\}$. Then $e_3 \in P$. Let $P_1$ be the subpath of $P$ between $s$ and the first node in common with $P_0$. Suppose first that this node is $v_2$. Since $P_0$ is a shortest $st$-path, it follows that $2 \leq |P_1| \leq 3$. By Claim 1, there is a 4-$st$-path $P'$ that does not contain $e_3$. Thus $P'$ intersects $\{e_1, e_2\}$. Let $P_1'$ be the subpath of $P'$ between $t$ and the first node in common with $P_0$. Note that $|P_1'| \leq 3$. If $P_1'$ contains another node of $P_1$ than $v_2$, say $z$, then the subpaths of $P_1$ between $s$ and $z$, and of $P_1'$ between $z$ and $t$ form a 4-$st$-path edge-disjoint from $P_0$, a contradiction. So, $P_1$ and $P_1'$ may only intersect in $v_2$. Moreover, if this is the case, since $P'$ must contain $e_1$ or $e_2$, we then have that $1 \leq |P_1'| \leq 2$. But this creates two edge-disjoint 4-$st$-paths, namely $P_1' \cup \{e_1, e_2\}$ and $P$, which is impossible. If $P_1$ and $P_1'$ do not intersect in a node, we have that $P_1'$ goes from $t$ to $v_1$. But therefore, the paths $P_1' \cup \{e_1\}$ and $P_1 \cup \{e_3\}$ are of length at most 4 and edge-disjoint, a contradiction.
Suppose now that the first node common to $P_1$ and $P_0$ is $v_1$. Then $P$ is either of the form $(P_1, v_1v_2, e_3)$, with $v_1v_2$ parallel to $e_2$ and $1 \leq |P_1| \leq 2$, or of the form $(sv_1, v_1u, uv_2, e_3)$ with $sv_1$ parallel to $e_1$ and $u \in N\backslash\{s, t, v_1, v_2\}$.

**Case 1:** $P = P_1 \cup \{v_1v_2, e_3\}$ where $v_1v_2$ is an edge of $[v_1, v_2]\backslash\{e_2\}$. By Claim 1, there is a 4-$st$-path not containing $e_3$. But it is not hard to see here that there are two edge-disjoint 4-$st$-paths, a contradiction.

**Case 2:** $P = (sv_1, v_1u, uv_2, e_3)$ with $sv_1 \in [s, v_1]\backslash\{e_1\}$ and $u \in N\backslash\{s, t, v_1, v_2\}$. Again, by Claim 1, there is a 4-$st$-path, say $P_2$, not containing $e_3$. Since $F$ is not a solution to the problem, $P_2$ must intersect all the 4-$st$-paths obtained from $P_0 \cup P_1$.

However, this is impossible without creating edge-disjoint 4-$st$-paths.

Therefore, $P$ must intersect the set $\{e_1, e_2\}$. By symmetry, $P$ must also intersect $\{e_2, e_3\}$. Now, to complete the proof of the claim, it suffices to show that $P$ also intersects $\{e_1, e_3\}$. Suppose the contrary. Then $P$ contains $e_2$, and hence $P = P_1 \cup \{e_2\} \cup P_2$, where $P_1$ is a path going from $s$ to $v_1$ and $P_2$ is a path going from $v_2$ to $t$. (The other case would immediately create two edge-disjoint 4-$st$-paths, namely, $P_1 \cup \{e_3\}$ and $\{e_1\} \cup P_2$). Note that $P_1$ and $P_2$ must be either both of length 1, or one of length 1 and the other of length 2. In both cases, by considering a 4-$st$-path not containing $e_2$, but intersecting all the previous paths, one would contradict the infeasibility of $F$.♦

Consider now the subgraph $G'_F$ of $G_F$ obtained by deleting the three edges of $P_0$. Since $F$ is not feasible, we have that $G'_F$ does not contain any 4-$st$-path. Thus, if $P'_0$ is a shortest $st$-path in $G'_F$, we have that $|P'_0| \geq 5$. Suppose first that $|P'_0| \geq 6$. We will show that there exists an inequality (6.7), with all the $W_i$'s empty, violated by $x^F$.

In $G'_F$, let $\Pi = (V_0, V_1, \ldots, V_6)$ be a partition of $N$ such that $V_i$, $i = 1, \ldots, 5$ contains the nodes at distance $i$ from $s$, and $V_6$ all the other nodes. Clearly, $s \in V_0$, and by our current assumption, $t \in V_6$. Moreover, we claim that each other $V_i$ is nonempty. Suppose this is not the case, that is, there is some $i \in \{1, \ldots, 5\}$ such that $V_i = \emptyset$. By definition, this means that there does not exist any node at distance $i$ from $s$, and hence, at distance 5 from $s$. Therefore $V_5 = \emptyset$ and the $st$-cut $\delta(V_6)$ is empty in $G'_F$. However, in $G_F$, we have by hypothesis that $x^F(\delta(V_6)) \geq 2$. As any $st$-path intersects any $st$-cut an odd number of times, we obtain that $e_1, e_2, e_3$ must all belong to $\delta(V_6)$. Therefore, $v_1 \in V_6$ and, as by Claim 4 there cannot exist a 4-$st$-path containing only one edge from $P_0$, we have that $v_2 \in V_4$. This also yields that all the $V_i$'s, except $V_5$, are nonempty. Consider now the 4-path-cut $T$ obtained from $\Pi$ by collapsing $V_5$ and $V_6$. In $G_F$, it is clear that the only chord of $T$ is $e_1$, and hence, $x^F(T) = 1$, which is again a contradiction.

Therefore, all the $V_i$'s are nonempty, and $\Pi$ is an admissible partition for a two-layered 4-path-cut inequality $ax \geq 4$ with the $W_i$'s empty. Moreover, in $G'_F$, we have $ax^F = 0$ by construction. Observe that we can suppose that $V_6$ only contains $t$. If not, we can put all the other nodes of $V_6$ in $V_5$ without creating a chord. Now consider this two-layered 4-path-cut in $G_F$. If $a(e_1) + a(e_2) + a(e_3) \leq 3$, we have that $ax^F < 4$ in $G_F$ and we have then found a violated inequality of type (6.7). Thus,

$a(e_1) + a(e_2) + a(e_3) \geq 4$. Since by Claim 4 no 4-$st$-path containing only $e_3$ from $P_0$ can exist in $G_F$, we have that $v_2 \in V_4 \cup V_5$. Suppose first that $v_2 \in V_4$. Thus $a(e_3) = 1$ and hence $a(e_1) + a(e_2) \geq 3$. The only possibility is that $v_1$ belongs to $V_6$. But this is impossible since $V_6$ only contains $t$. Suppose now that $v_2 \in V_5$. Therefore $a(e_3) = 0$ and $a(e_1) + a(e_2) \geq 4$. Clearly, this is also impossible.

Finally, suppose that the shortest $st$-path in $G'_F$ is of length exactly 5. We claim that there exists an inequality (6.7), with at least one $W_i$ nonempty, violated by $x^F$.

By Claim 1, there exists in $G_F$ a 4-$st$-path $P_i$ not containing $e_i$, for each $i = 1, 2, 3$. Moreover, by Claim 4, the $P_i$'s must contain $P_0 \setminus \{e_i\}$. Besides these two edges from $P_0$, it is clear that each $P_i$ must contain two more edges. Indeed, if one $P_i$ was of length 3, this would create two edge-disjoint 4-$st$-paths in $F$, a contradiction. For the same reason, the $P_i$'s cannot have a node in common besides $s, v_1, v_2, t$. As a consequence, we have the graph of Figure 6.2 as a subgraph of $G_F$. Note that the $st$-path of this subgraph not intersecting $P_0$ is of length 6. Let us denote its nodes by $s, u_1, v_1, u_2, v_2, u_3, t$. Remark that, as the shortest $st$-path of $G'_F$ is of length exactly 5, there must be in $G_F$ additional edges (and nodes) forming, eventually with edges of that path, an $st$-path of length 5.

Consider the partition $\Pi = (V_0, \ldots, V_6, W_1, \ldots, W_4)$ in $G_F$ defined as follows. We set $V_0 = \{s\}$, $V_1 = \{u_1\}$, $V_2 = \{v_1\}$, $V_3 = \{u_2\}$, $V_4 = \{v_2\}$, $V_5 = \{u_3\}$, and $V_6 = \{t\}$. All the other nodes are distributed to the $W_i's$ through a breadth first search from $s$ in $G_F$. See Figure 6.5 for an illustration. Note that some $W_i$'s may be empty, but not all of them by our previous remark. Let $\overline{E} = (\bigcup_{j-i\geq 2 \text{ or } i-j\geq 3}[V_i, W_j]) \cup (\bigcup_{(i,j)=(2,3),(3,1),(3,4),(4,2)}[V_i, W_j])$, that is, the set of edges between the two layers of the partition $\Pi$ that have a positive coefficient in the corresponding inequality (6.7). We claim that $G_F$ does not contain any edge from $\overline{E}$. Suppose this is not the case. If there was an edge of $G_F$ in $[V_i, W_j]$ with $j - i \geq 2$, then its end node in $W_j$ would be at distance $i + 1$ from $s$, and hence, it should have been put in $W_{i+1}$ by construction. The same contradiction holds for edges of $G_F$ in $[V_2, W_3] \cup [V_3, W_4]$. Now, assume there is an edge $e$ of $G_F$ in $[V_i, W_j]$ with $i - j \geq 3$. Thus, by construction, there exists in $G_F$ a subpath $P$, from $s$ to the end node of $e$ in $W_j$, of length exactly $j$. But then, $G_F$ contains the graph of Figure 6.2, $P$ and $e$, and hence, two edge-disjoint 4-$st$-paths, which is impossible. Finally, if $G_F$ uses some edge from $[V_3, W_1] \cup [V_4, W_2]$, we get a similar contradiction.

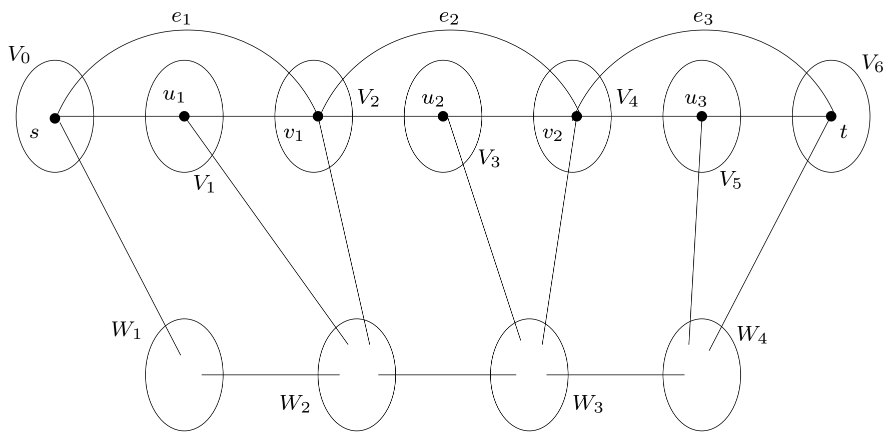


Figure 6.5: Possible edges in the constructed partition $\Pi$

Consequently, the only edges of $G_F$ between the two layers of partition $\Pi$ have coefficient zero in the corresponding inequality (6.7). Moreover, by construction, the lower partition cannot contain chords, and the upper one has for only chords $e_1, e_2, e_3$. Since these three edges have a coefficient 1, we obtain that $ax^F = 3 < 4$, and the proof is complete. □

## 6.4 Concluding remarks

In this chapter, we have considered the Two edge-disjoint Hop-constrained Paths Problem, along with the node-disjoint version of this problem, when $L = 4$. For both versions, we have presented new families of valid inequalities, and obtained an integer programming formulation in the space of the design variables. These theoretical results can also be useful in practice, for example for verifying whether a given solution is feasible or not for the THPP with $L = 4$. From the proofs of Theorems 6.2.1 and 6.3.2, one could indeed derive separation procedures for an integer solution relatively to the constraints of the formulations, as stated by the following theorem.

**Theorem 6.4.1.** *Given an integer solution $x^*$, the problem of finding whether or not $x^*$ satisfies system (6.1)-(6.6) ((6.1),(6.3),(6.5),(6.6),(6.7)) can be solved in polynomial time.*

Theorem 6.4.1 is very important from a practical point of view. Indeed, in many approaches like cutting planes approaches, one has to solve the feasability problem for a given (integer) solution. If the solution is feasible for the underlying problem, then it is optimal. By Theorem 6.4.1, this problem can be solved in polynomial time for both the node and edge THPP.

Of course, it would be interesting to know separation procedures for these inequalities relatively to any solution (integer or not). Indeed, if the $st$-cut (and $st$-node-cut) inequalities can always be separated exactly in polynomial time, it is already not the case for the $L$-$st$-path-cut (and $L$-$st$-path-node-cut) inequalities. More precisely, in [25], Fortz et al. propose a polynomial-time separation of these inequalities when $L = 2, 3$, and show that it becomes $NP$-hard for $L \geq 12$. In particular, the complexity of the separation problem associated to the 4-$st$-path-cut inequalities remains an open question (see [56]). Moreover, we still need to investigate how to separate the two-layered 4-path-cut inequalities.

Since the THPP for $L = 4$ can be solved in polynomial time by enumeration (at least in simple graphs), from the equivalence between optimization and separation, it follows that inequalities (6.3) and (6.4), as well as inequalities (6.7), can be separated in polynomial time among a system of inequalities describing the THPP polytope in that case.

Also, a natural question that may be posed is whether or not these formulations are complete, that is whether or not their linear relaxation is integral. Unfortunately, for the node version, this is not the case as shown by the following example.

Consider the graph $G = K_6$ of Figure 6.6, where the edges in solid lines have value $1/2$, the ones in bold have value 1, and the remaining edges have value zero. It is easy to verify that this solution is a fractional extreme point of the polyhedron given by the linear relaxation of the node THPP with $L = 4$ and $D = \{\{1, 6\}\}$. This point can be cut off by the following valid inequality

$$2x(e_2) + x(e_3) + x(e_4) + 2x(e_5) + x(e_6) + x(e_7) + 2x(e_9) + x(e_{11}) + x(e_{14}) + 2x(e_{15}) \geq 3.$$

Moreover, this inequality is facet-defining for the polytope on this graph.

An interesting question would be to see whether the linear relaxation of the edge version is integral.

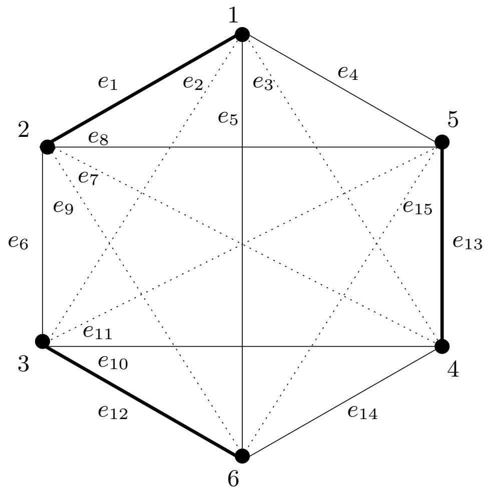


Figure 6.6: Fractional extreme point for the node THPP with $L = 4$

Finally, we would like to obtain such an integer programming formulation for any greater values of $L$, starting from $L = 5$. Note that, from our experience regarding $L = 4$, it seems that the node THPP is simpler than the edge THPP for that matter. A starting point would be, for example, to find how to cut off the infeasible solution of Figure 6.1. This work could also be extended to the more general problems discussed in the previous chapters, that is, when more than two paths are required or when several demands have to be linked. For example, the formulations given in this chapter can be easily generalized to the case where more than one pair of terminals is considered. Here, for each version, the formulation is given by the inequalities related to every pair $\{s, t\}$ of terminals, together with the integrality constraints. Hence, an efficient separation algorithm for inequalities (6.3), (6.4) and (6.7) would be of great interest for solving the multiple demands case by cutting planes.

# Conclusion

In this thesis, we considered the $k$-edge connected $L$-hop-constrained network design problem. More precisely, we studied several particular cases of this problem according to the values taken by the parameters $k$ and $L$, along with the kind of demand pairs (single, rooted, arbitrary). We also considered the node connected case.

In the case of a single demand $\{s,t\} \in D$, we were able to obtain a complete and minimal description of the associated polytope for any $k \geq 2$ when $L = 2$, and for $k = 2$ when $L = 3$. Moreover, since all the underlying inequalities ($st$-cut and $L$-$st$-path-cut inequalities) are separable in polynomial time when $L = 2, 3$, we devised an effective cutting plane algorithm to solve the problem in that case. This one was indeed a lot better in practice than the enumerative procedure. The computational results also appeared to confirm that the previous description is still complete for any $k \geq 2$ when $L = 3$ (Conjecture 2.8.1). In addition, we noticed that the problem gets more complicated when $k$ is about half the number of nodes in the graph.

When the set of demands is not reduced to a single pair, we have shown that the problem is strongly $NP$-hard, for $k = 2$ and all fixed value of $L \geq 2$, even when these demands have one root node in common. If the $st$-cut and $L$-$st$-path-cut inequalities (written for all $\{s,t\} \in D$) no longer suffice to completely describe the associated polytope, they still constitute an integer programming formulation of the problem when $L = 2, 3$. Also, we gave several new classes of valid inequalities, adapted first to rooted demands, and then to disjoint ones. For the former ones, we studied necessary and sufficient conditions for them to be facet-defining. Finally, we embedded separation procedures for those different classes into the framework of a Branch-and-Cut algorithm, and presented extensive computational results for both random and real instances.

From this experimental study, it appeared that the problem for $L = 3$ is a lot more difficult to solve in practice than that for $L = 2$. However, the additional classes of valid inequalities that we introduced were more effective in the former case than in the latter one, where only separating the constraints of the formulation was often satisfactory. In particular, the best improvements of the gaps, thanks to those new inequalities, were made for real instances with $L = 3$.

Finally, we came back to the case where there is only a single demand and $k = 2$, since at first we were not able to find an integer programming formulation, in the space of the design variables, when $L \geq 4$. After some work, we obtained several new classes of valid constraints for the problem, not only when the two required $L$-$st$-paths must be edge-disjoint, but also when they need to be node-disjoint. In each case, we then gave such a formulation when $L = 4$. Note that this one can be easily extended to a non-singleton set $D$ by writing together the constraints for each demand. Unfortunately, in the node-disjoint case, the proposed formulation appeared to be neither sufficient for $L \geq 5$, nor complete to describe the associated polytope for $L = 4$. These two questions also remain open in the edge-disjoint case.

In the future, it would be interesting to pursue this study in the following directions. As already mentioned, Conjecture 2.8.1 remains to be proved regarding $L = 3$. On the other hand, we still lack a natural formulation for all values of $L \geq 5$. Actually, for $L = 4$, we would like to know if the natural formulation we found completely describes the associated polytope or not. It would also be interesting to know how to separate the so-called two-layered 4-path-cut inequalities, and how to extend them to any $L$.

In the case of multiple demands, some additional work needs to be done, especially in the case of arbitrary pairs. Our Branch-and-Cut algorithm could indeed be improved to always give better results. When $L = 3$, the final gaps observed after 5 hours are quite large for the biggest random instances. If some improvement could be made regarding the lower bounds by devising new facet-defining inequalities to separate, the construction of good heuristical solutions for the upper bounds also needs to be addressed.

# Bibliography


[1] M. Baïou and A. R. Mahjoub. Steiner 2-edge connected subgraph polytopes on series-parallel graphs. *SIAM J. Discrete Math.*, 10(3):505–514, 1997.

[2] A. Balakrishnan and K. Altinkemer. Using a hop-constrained model to generate alternative communication network design. *ORSA J. Comput.*, 4(2):192–205, 1992.

[3] F. Barahona and A. R. Mahjoub. On two-connected subgraphs polytopes. *Discrete Math.*, 147(1-3):19–34, 1995.

[4] W. Ben-Ameur. Constrained length connectivity and survivable networks. *Networks*, 36(1):17–33, 2000.

[5] W. Ben-Ameur and E. Gourdin. Internet routing and related topology issues. *SIAM J. Discrete Math.*, 17:18–49, 2003.

[6] M. Didi Biha and A. R. Mahjoub. $k$-edge connected polyhedra on series-parallel graphs. *Oper. Res. Lett.*, 19:71–78, 1996.

[7] M. Didi Biha and A. R. Mahjoub. Steiner $k$-edge connected subgraph polytope. *J. Comb. Optim.*, 4(1):131–144, 2000.

[8] A. Bley. On the complexity of vertex-disjoint length-restricted path problems. Preprint SC-98-20, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1998. To appear in Computational Complexity.

[9] J. A. Bondy and U. S. R. Murty. *Graph theory with applications.* Universities Press, Belfast, 1976.

[10] S. C. Boyd and T. Hao. An integer polytope related to the design of survivable communication networks. *SIAM J. Discrete Math.*, 6(4):612–630, 1993.

[11] J. Christophe, S. Dewez, J.-P. Doignon, S. Elloumi, G. Fasbender, P. Grégoire, D. Huygens, M. Labbé, H. Mélot, and H. Yaman. Linear inequalities among graph invariants: using graphedron to uncover optimal relationships, 2004. Submitted.

[12] C. R. Coullard, A. B. Gamble, and J. Liu. The $k$-walk polyhedron. *Advances in Optimization and Approximation, Nonconvex Optim. Appl. 1, D.-Z. Du and J. Sun, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands*, pages 9–29, 1994.

[13] G. Dahl. The 2-hop spanning tree problem. *Oper. Res. Lett.*, 23(1-2):21–26, 1998.

[14] G. Dahl. Notes on polyhedra associated with hop-constrained paths. *Oper. Res. Lett.*, 25(2):97–100, 1999.

[15] G. Dahl, N. Foldnes, and L. Gouveia. A note on hop-constrained walk polytopes. *Oper. Res. Lett.*, 32:345–349, 2004.

[16] G. Dahl and L. Gouveia. On the directed hop-constrained shortest path problem. *Oper. Res. Lett.*, 32:15–22, 2004.

[17] G. Dahl and L. Gouveia. On formulations of the hop-constrained minimum spanning tree problem, preprint, 2001.

[18] G. Dahl, D. Huygens, A. R. Mahjoub, and P. Pesneau. On the $k$ edge-disjoint 2-hop-constrained paths polytope, 2004. Submitted.

[19] G. Dahl and B. Johannessen. The 2-path network problem. *Networks*, 43(3):190–199, 2004.

[20] G. Exoo. A family of graphs and the degree/diameter problem. *J. of Graph Theory*, 37(2):118–124, 2001.

[21] J. Fonlupt and A. R. Mahjoub. Critical extreme points of the 2-edge connected spanning subgaph polytope. *Integer programming and combinatorial optimization, Lecture Notes in Comput. Sci. 1610, G. Cornuéjols, R. E. Burkard, and G. J. Woeginger, eds., Springer-Verlag, Berlin*, pages 166–182, 1999.

[22] B. Fortz and M. Labbé. Polyhedral results for two-connected networks with bounded rings. *Math. Program.*, 93(1):27–54, 2002.

[23] B. Fortz and M. Labbé. Two-connected networks with rings of bounded cardinality. *Computational Optimization and Applications*, 27(2):123–148, 2004.

[24] B. Fortz, M. Labbé, and F. Maffioli. Solving the two-connected network with bounded meshes problem. *Oper. Res.*, 48(6):866–877, 2000.

[25] B. Fortz, A. R. Mahjoub, S. T. McCormick, and P. Pesneau. The 2-edge connected subgraph problem with bounded rings, 2003. *Math. Prog.*, to appear.

[26] H. N. Gabow, M. X. Goemans, and D. P. Williamson. An efficient approximation algorithm for the survivable network design problem. Networks and matroids; Sequencing and scheduling. *Math. Programming*, 82(1-2):13–40, 1998.

[27] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness.* Freeman, New York, 1979.

[28] M. X. Goemans and V. S. Ramakrishnan. Minimizing submodular functions over families of sets. *Combinatorica*, 15:499–513, 1995.

[29] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *SIAM J. Appl. Math.*, 9(4):551–570, 1961.

[30] L. Gouveia. Multicommodity flow models for spanning trees with hop constraints. *European J. Oper. Res.*, 95(1):178–190, 1996.

[31] L. Gouveia. Using variable redefinition for computing lower bounds for minimum spanning and Steiner trees with hop constraints. *INFORMS J. Comput.*, 10(2):180–188, 1998.

[32] L. Gouveia and E. Janssen. Designing reliable tree networks with two cable technologies. *European J. Oper. Res.*, 105(3):552–568, 1998.

[33] L. Gouveia and T. L. Magnanti. Modelling and solving the diameter-constrained minimum spanning tree problem, 2000. Technical report, DEIO-CIO, Faculdade de Ciências da Universidade de Lisboa, Portugal.

[34] L. Gouveia and C. Requejo. A new Lagrangean relaxation approach for the hop-constrained minimum spanning tree problem. *European J. Oper. Res.*, 132(3):539–552, 2001.

[35] M. Grötschel, L. Lovàsz, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[36] M. Grötschel and C. L. Monma. Integer polyhedra arising from certain networks design problems with connectivity constraints. *SIAM J. Discrete Math.*, 3(4):502–523, 1990.

[37] M. Grötschel, C. L. Monma, and M. Stoer. Polyhedral approaches to network survivability. *Reliability of Computer and Communication Networks, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 5, F. Roberts, F. Hwang, and C. L. Monma, eds., AMS, Providence, RI*, 5:121–141, 1991.

[38] M. Grötschel, C. L. Monma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Oper. Res.*, 40(2), 1992.

[39] M. Grötschel, C. L. Monma, and M. Stoer. Facets for polyhedra arising in the design of communication networks with low-connectivity constraints. *SIAM J. Optim.*, 2(3):474–504, 1992.

[40] M. Grötschel, C. L. Monma, and M. Stoer. Design of survivable networks. *Network Models, Handbooks Oper. Res. Management Sci. 7, M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemahauser, eds., North-Holland, Amsterdam*, 7:617–671, 1995.

[41] M. Grötschel, C. L. Monma, and M. Stoer. Polyhedral and computational investigations for designing communication networks with high survivability requirements. *Oper. Res.*, 43(6), 1995.

[42] M. Grötschel and M. W. Padberg. Polyhedral theory. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan and D. B. Shmoys, eds., Wiley, New-York*, 1985.

[43] D. Gusfield. Very simple methods for all pairs network flow analysis. *SIAM J. Comput.*, 19(1):143–155, 1990.

[44] J. Hao and J. B. Orlin. A faster algorithm for finding the minimum cut in a directed graph. *J. Algorithms*, 17(3):424–446, 1994.

[45] D. Huygens, M. Labbé, A. R. Mahjoub, and P. Pesneau. The two-edge connected hop-constrained network design problem : Valid inequalities and branch & cut, 2005. Submitted.

[46] D. Huygens and A. R. Mahjoub. Integer programming formulations for the two 4-hop-constrained paths problem, 2005. Submitted.

[47] D. Huygens, A. R. Mahjoub, and P. Pesneau. Two edge-disjoint hop-constrained paths and polyhedra. *SIAM J. Discrete Math.*, 18(2):287–312, 2004.

[48] A. Itai, Y. Perl, and Y. Shiloach. The complexity of finding maximum disjoint paths with length constraints. *Networks*, 12:277–286, 1982.

[49] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21:39–60, 2001.

[50] H. Kerivin and A. R. Mahjoub. Design of survivable networks: A survey, 2005. *Networks*, to appear.

[51] H. Kerivin, A. R. Mahjoub, and C. Nocq. $(1,2)$-survivable networks: facets and branch-and-cut. *The Sharpest Cut : The Impact of Manfred Padberg and His Work, MPS/SIAM Ser. Optim., M. Grötschel, ed., SIAM, Philadelphia*, pages 121–152, 2004.

[52] L. J. Leblanc and R. Reddoch. Reliable link topology/capacity and routing in backbone telecommunication networks. Working paper 90-08, Owen Graduate School of Management, Vanderbilt University, Nashville, TN, 1990.

[53] L. J. Leblanc, R. Reddoch, J. Chifflet, and P. Mahey. Packet routing in telecommunications networks with path and flow restrictions. *INFORMS J. Comput.*, 11(3):188–197, 1999.

[54] C.-L. Li, S. T. McCormick, and D. Simchi-Levi. Finding disjoint paths with different path-costs: Complexity and algorithms. *Networks*, 22(7):653–667, 1992.

[55] A. R. Mahjoub. Two-edge connected spanning subgraphs and polyhedra. *Math. Programming*, 64A(2):199–208, 1994.

[56] A. R. Mahjoub and S. T. McCormick. The complexity of max flow and min cut with bounded-length paths, 2004. Preprint.

[57] B. D. McKay, M. Miller, and J. Siran. A note on large graphs of diameter two and given maximum degree. *J. on Comb. Theory Ser. B*, 74(1):110–118, 1998.

[58] C. L. Monma and D. F. Shallcross. Methods for designing communications networks with certain two-connected survivability constraints. *Oper. Res.*, 37(4):531–541, 1989.

[59] H. Pirkul and S. Soni. New formulations and solution procedures for the hop constrained network design problem. *European J. Oper. Res.*, 148(1):126–140, 2003.

[60] W. R. Pulleyblank. Polyhedral combinatorics. *Optimization, Handbooks Oper. Res. Management Sci. 1, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, eds., North-Holland, Amsterdam*, 1:371–446, 1989.

[61] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency.* Algorithms and Combinatorics, Vol. 24. Springer, Berlin, Heidelberg, 2003.