Available online at www.sciencedirect.com
SCIENCE DIRECT®

Discrete Applied Mathematics 133 (2004) 103–121

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

Bidirected and unidirected capacity installation in telecommunication networks

Stan P.M. van Hoesel[a,*], Arie M.C.A. Koster[b,1], Robert L.M.J. van de Leensel[b,1], Martin W.P. Savelsbergh[c]

[a]*Department of Quantitative Economics, Faculty of Economics & Business Admn., University of Maastricht, P.O. Box 616, Maastricht, 6200 MD, Netherlands*
[b]*Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustraße 7, D-14195 Berlin, Germany*
[c]*School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, 30332-0205 GA, USA*

Received 19 October 2000; received in revised form 29 October 2001; accepted 15 December 2002

Abstract

In the design of telecommunication networks, decisions concerning capacity installation and routing of commodities have to be taken simultaneously. Network Loading problems formalize these decisions in mathematical optimization models. Several variants of the problem exist: bifurcated or non-bifurcated routing, bidirected or unidirected capacity installation, and symmetric versus non-symmetric routing restrictions. Moreover, different concepts of reliability can be considered. In this paper, we study the polyhedral structure of two basic problems for non-bifurcated routing: network loading with bidirected and unidirected capacity installation.

We show that strong valid inequalities for the substructure restricted to a single edge, are also strong valid inequalities for the overall models. In a computational study, several classes of inequalities, both for the substructure and the overall problem, are compared on real-life instances for both variants of network loading.

Keywords: Network loading; Telecommunication network design; Polyhedral combinatorics; Branch-and-cut

* Corresponding author. Tel.: +31(0)43-388-3727; fax: +31(0)43-388-4874.
E-mail addresses: s.vanhoesel@ke.unimaas.nl (S.P.M. van Hoesel), koster@zib.de (A.M.C.A. Koster), rvandeleensel@yahoo.com (R.L.M.J. van de Leensel), mwps@isye.gatech.edu (M.W.P. Savelsbergh).
URL: http://www.zib.de/koster/

[1] The research was partly carried out while these authors were Ph.D. students at Maastricht University.

doi:10.1016/S0166-218X(03)00436-0

1. Introduction

The network design or network loading problem (NLP) occurs in telecommunication networks, where demand for capacity of multiple commodities is to be realized by inserting capacity into a given network. The capacity can be placed in different sizes, usually multiples of each other. We restrict ourselves to a single capacity size, although many of the ideas presented in this paper can be extended in case multiple capacity sizes are available. Along with a capacity plan, a routing of all commodities is to be determined. The capacity plan should suffice to accommodate the demands of all commodities simultaneously on the given routings. This problem has been studied in many variants with respect to network lay-out, capacity usage, and routing possibilities. Routing of the demand can be done by reserving capacity on a subnetwork that consists of a path between the endpoints of a commodity only (non-bifurcated routing), or of a set of paths (bifurcated routing). We only consider the non-bifurcated routing. This case has also been studied by Gavish and Altinkemer [11] and Brockmüller et al. [8,9]. For the bifurcated case we refer to Magnanti et al. [14,15]. With respect to capacity usage one can distinguish unidirectional and bidirectional capacity usage, i.e., if an edge contains a unit of capacity, this unit can either be used in one or in both directions of the edge. In most studies the unidirected case is examined. Bienstock and Günlük [7] and Bienstock et al. [6], however, study the bidirected case. In this paper, we consider both forms of capacity usage. Depending on the network technology (protocol), network operators often require symmetric routing. This restriction does not fundamentally change the models (see Section 2), and is therefore not discussed here. Moreover, we show that the models of the corresponding NLPs have many common aspects. To emphasize these common properties, no further application-specific design constraints are incorporated. For instance, we do not take reliability requirements into account. For the design of survivable networks with bifurcated routing we refer to Wessäly [18] and the references therein, for non-bifurcated routing to Van de Leensel [13].

In this paper, we study the equivalences and differences between the models for bidirected and unidirected capacity installation. In Section 2, we formally describe these $\mathcal{NP}$-hard problems, and discuss path and flow formulations for both problems. Next, in Section 3, we start our polyhedral investigations of the models by determining the dimension of each of the formulations. In all formulations a similar class of constraints occurs: the edge capacity constraints. In Section 4, we prove the strong relationship between valid inequalities for the network loading problems and the polytope defined by a single edge constraint.

The polyhedral structure of the single edge polytope (with bidirected and unidirected capacity usage) is studied in Section 5. This section also includes an overview of inequalities for the overall problems. The computational results achieved on real-life instances with a branch-and-cut algorithm can be found in Section 6. This includes a comparison of the effectiveness of the inequalities.

2. Problem description and formulations

Let $G=(V,E)$ be an undirected connected graph with node set V and edge set E. Given the graph G we define the arc set A, which contains two directed arcs (i,j) and (j,i) for all edges $e=\{i,j\}\in E$. Let Q be a set of demands (commodities). Each element $q\in Q$ is a triple (s^q,t^q,d^q), with $s^q,t^q\in V$, $s^q\neq t^q$, representing a commodity with positive integer demand size $d^q\in\mathbb{Z}^+$ that must be routed from source node s^q to sink node t^q on a single path through the network. To route a set of commodities on an arc, sufficient capacity must be available on the corresponding edge. The capacity on an edge is determined by the number of capacity units installed on the edge, where each unit has a base capacity $\lambda\in\mathbb{Z}^+$. Either bidirected or unidirected capacity can be installed. The installation of a unit of bidirected capacity implies that the capacity can be used twice, once in each direction. In case unidirected capacity is installed, the capacity is available only once, and have to be shared by commodities in both directions. The goal is to minimize the costs of the installed capacity in the network while ensuring that all commodities can be routed from source to sink simultaneously.

Depending on the telecommunication protocol, symmetric routing can be required. This means that a commodity q with source s^q and sink t^q has to be routed via the same (undirected) path as a commodity q' with source $s^{q'}=t^q$ and sink $t^{q'}=s^q$. For unidirectional capacity, this implies that the two commodities reduce to a single commodity with demand $d^q+d^{q'}$. In case of bidirected capacity, the symmetric routing requirement results in a slightly different model. Because of this minor impact of the symmetric routing requirement on the models, we leave it out of the remaining discussion.

Both the bidirected and the unidirected version of the studied network loading problem are $\mathcal{NP}$-hard [13]. We assume that for each commodity $q\in Q$ there exist at least two node-disjoint paths from source node to sink node (node-disjoint, except for the nodes s^q and t^q). If this assumption is not satisfied, the graph G contains a separating vertex, hence the problem can be decomposed into smaller problems that do satisfy the assumption. Next, we present for both the unidirected and bidirected case a flow and path formulation of the model.

2.1. The unidirected non-bifurcated flow model (UNFM)

To formulate this problem as an integer program, let $x_{ij}\in\mathbb{Z}_0^+$ be the number of capacity units installed on edge $\{i,j\}$, and let f_{ij}^q be a binary variable indicating whether the commodity $q\in Q$ is routed via arc $(i,j)\in A$ or not. If c_{ij} represents the costs per base capacity unit on edge $\{i,j\}\in E$, then the model reads

$$\min \quad \sum_{\{i,j\}\in E} c_{ij}x_{ij} \tag{1}$$

$$\text{s.t.}\quad \sum_j f_{ij}^q-\sum_j f_{ji}^q=\begin{cases}1 & \text{if } i=s^q,\\ -1 & \text{if } i=t^q, \quad \forall q\in Q,\ \forall i\in V,\\ 0 & \text{otherwise,}\end{cases} \tag{2}$$

$$\lambda x_{ij} \geqslant \sum_{q \in Q} d^q (f_{ij}^q + f_{ji}^q) \qquad \forall \{i,j\} \in E, \tag{3}$$

$$f_{ij}^q, f_{ji}^q \in \{0,1\},\ x_{ij} \in \mathbb{Z}_0^+ \qquad \forall q \in Q,\ \forall \{i,j\} \in E. \tag{4}$$

This model is called the unidirected non-bifurcated flow model, and the corresponding set of feasible solutions is denoted *UNFM*. The capacity on an edge is *unidirected* because installed capacity can be used by traffic in both directions, i.e., the required capacity on an edge is determined by the sum of forward and backward flow on the edge. It is called *non-bifurcated* since the demand of a commodity has to be routed on a single path (i.e. the demand cannot be bifurcated). Finally, *flow* variables on individual arcs are used to model the routing of a commodity from source node to sink node. Note that, bifurcated routing can be obtained by relaxation of the binary constraints of the flow variables.

2.2. The unidirected non-bifurcated path model (UNPM)

Instead of using flow variables on individual edges to model routing restrictions, one can also use binary variables z_p^q representing whether a certain path $p \in P^q$ (the set of all possible paths for the commodity q) is used to route the commodity q from source node s^q to sink node t^q. We assume that P^q only contains simple paths, that is paths that visit each node at most once. If $P_{ij}^q \subseteq P^q$ denotes the set of paths for commodity q that contain arc (i,j), then this leads to the following unidirected non-bifurcated path model *UNPM*:

$$\min \quad \sum_{\{i,j\} \in E} c_{ij} x_{ij} \tag{5}$$

$$\text{s.t.} \quad \sum_{p \in P^q} z_p^q = 1 \qquad \forall q \in Q, \tag{6}$$

$$\lambda x_{ij} \geqslant \sum_{q \in Q} \sum_{p \in P_{ij}^q \cup P_{ji}^q} d^q z_p^q \qquad \forall \{i,j\} \in E, \tag{7}$$

$$z_p^q \in \{0,1\},\ x_{ij} \in \mathbb{Z}_0^+ \qquad \forall q \in Q,\ \forall p \in P^q,\ \forall \{i,j\} \in E. \tag{8}$$

Again, the bifurcated case can be dealt with by relaxing the binary constraints, this time for the z_p^q variables.

2.3. The bidirected non-bifurcated flow model (BNFM)

Depending on the exact application and level of aggregation, capacity that is installed on edges in the network can also be *bidirected*, i.e. each unit of capacity installed on an edge $\{i,j\}$ gives a capacity of λ on both corresponding arcs (i,j) and (j,i), and capacity consumption is bidirected as well. This leads to the following bidirected non-bifurcated flow model *BNFM*, with feasible solution set

$$\min \quad \sum_{\{i,j\}\in E} c_{ij}x_{ij} \tag{9}$$

$$\text{s.t.} \quad \sum_j f_{ij}^q - \sum_j f_{ji}^q = \begin{cases} 1 & \text{if } i = s^q, \\ -1 & \text{if } i = t^q \ \ \forall q\in Q, \ \forall i \in V, \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

$$\lambda x_{ij} \geqslant \sum_{q\in Q} d^q f_{ij}^q \qquad \forall \{i,j\}\in E, \tag{11}$$

$$\lambda x_{ij} \geqslant \sum_{q\in Q} d^q f_{ji}^q \qquad \forall \{i,j\}\in E, \tag{12}$$

$$f_{ij}^q, f_{ji}^q \in \{0,1\}, \ x_{ij} \in \mathbb{Z}_0^+ \qquad \forall q\in Q, \ \forall \{i,j\}\in E. \tag{13}$$

2.4. The bidirected non-bifurcated path model (BNPM)

Similar to the unidirected case, one can model the bidirected case using path variables. This bidirected non-bifurcated path model *BNPM*, with feasible solution set reads

$$\min \quad \sum_{\{i,j\}\in E} c_{ij}x_{ij} \tag{14}$$

$$\text{s.t.} \quad \sum_{p\in P^q} z_p^q = 1 \qquad \forall q \in Q, \tag{15}$$

$$\lambda x_{ij} \geqslant \sum_{q\in Q} \sum_{p\in P_{ij}^q} d^q z_p^q \quad \forall \{i,j\}\in E, \tag{16}$$

$$\lambda x_{ij} \geqslant \sum_{q\in Q} \sum_{p\in P_{ji}^q} d^q z_p^q \quad \forall \{i,j\}\in E, \tag{17}$$

$$z_p^q \in \{0,1\}, \ x_{ij}\in \mathbb{Z}_0^+ \quad \forall q\in Q, \ \forall p \in P^q, \ \forall \{i,j\}\in E. \tag{18}$$

3. Dimension and trivial facets

In this paper we focus on the polyhedral structure of the polytopes defined by the convex hull of integer solutions of *UNFM*, *UNPM*, *BNFM*, and *BNPM*. To do so, the dimension of each of the polytopes is an important notion.

Proposition 1. *The dimension of both conv(UNPM) and conv(BNPM) is equal to* $|E| + \sum_{q\in Q}(|P^q| - 1)$.

Proof. The number of edge capacity variables equals $|E|$ and the number of path variables equals $\sum_{q\in Q}|P^q|$. Since the number of linearly independent equality constraints equals $|Q|$, this leads to an upper bound on the dimension of $|E|+\sum_{q\in Q}(|P^q|-1)$. Next, we state $1+|E|+\sum_{q\in Q}(|P^q|-1)$ affinely independent feasible solutions, which proves our claim. In the first solution each commodity $q\in Q$ is routed via an arbitrarily chosen path $\hat{p}\in P^q$, and the capacity equals the total flow on an edge rounded up to the nearest multiple of λ. Given this solution we can install an extra capacity unit on each edge, which yields another $|E|$ affinely independent solutions. Finally, for each commodity $q\in Q$ and each path $p\in P^q\setminus\{\hat{p}\}$ we construct a solution by keeping the routing of all other commodities fixed as in the first solution, but replacing path $\hat{p}$ by path p for commodity q, and installing additional capacity if needed. The $\sum_{q\in Q}(|P^q|-1)$ vectors that are obtained are affinely independent since each solution contains a path variable that is not used in any other solution vector. □

The following lemma indicates that the number of path variables in the formulation, and therefore the dimension of the corresponding polytope, can become exponentially large in terms of the size of the graph.

Lemma 2. *The number of distinct simple paths* (*a path without node repetition*) *between any pair of nodes in a complete graph on* $|V|$ *nodes equals* $\lfloor(|V|-2)!e\rfloor$, *if* $|V|\geqslant 3$.

Proposition 3. *The dimension of both conv*(*UNFM*) *and conv*(*BNFM*) *is equal to* $|E|+|Q|(|A|-|V|+1)$.

Proof. The number of edge capacity variables equals $|E|$ and the number of flow variables equals $|Q|\cdot|A|$. Furthermore, since for each commodity there are $|V|$ flow balance constraints, of which $|V|-1$ are linearly independent, an upper bound on the dimension is given by $|E|+|Q|(|A|-|V|+1)$. To prove that this bound is tight we show that there exist no other implicit equalities in the model. Stated differently, if

$$\sum_{\{i,j\}\in E}\alpha_{ij}x_{ij}+\sum_{q\in Q}\sum_{\{i,j\}\in E}(\beta_{ij}f_{ij}^q+\beta_{ji}f_{ji}^q)=\delta$$

is satisfied by each solution in *UNFM*, we prove that this equality is a linear combination of the model equalities.

Let $\{u,v\}\in E$, and let $(x,f)\in UNFM$. Next, define a solution $(\bar{x},\bar{f})$ as $\bar{f}=f$, $\bar{x}_{uv}=x_{uv}+1$ and $\bar{x}_{ij}=x_{ij}$ for all $\{i,j\}\neq\{u,v\}$. Because both solutions satisfy the equality, it holds that $\alpha_{uv}=0$, and since the edge was chosen arbitrarily it follows that $\alpha_{ij}=0$ for all $\{i,j\}\in E$.

Next we show that for all $q\in Q$ and for all cycles C in the graph it holds that $\sum_{(i,j)\in C}\beta_{ij}^q=0$. Since any cycle in the graph can be decomposed into a collection of simple cycles (i.e., cycles that visit each node at most once) it follows that we only have to prove this claim for simple cycles.

Let $\hat{q}\in Q$ and C a simple cycle in the graph. First we consider the case that C is a 2-cycle (a cycle of two arcs, say (u,v) and (v,u) for some $u,v\in V$). Since there exist two node disjoint paths from $s^{\hat{q}}$ to $t^{\hat{q}}$ in the graph, there exists a path from $s^{\hat{q}}$ to $t^{\hat{q}}$ that does not contain edge $\{u,v\}$. Let $(x,f) \in UNFM$ be a solution that uses this specific path for the routing of commodity $\hat{q}$. Given this solution, let $(\bar{x},\bar{f}) \in UNFM$ be a solution that employs exactly the same routing strategy for all commodities $q \in Q$, except that commodity $\hat{q}$ is additionally routed on arcs (u,v) and (v,u). Since both solutions satisfy the equality and $\alpha_{ij}=0$ for all $\{i,j\} \in E$ it follows that $\beta^{\hat{q}}_{uv}+\beta^{\hat{q}}_{vu}=0$.

Now we consider the case that C is not a 2-cycle. Let p be a simple path from $s^{\hat{q}}$ to $t^{\hat{q}}$ in the graph. If the number of nodes on the path p that are also on the cycle C is less than or equal to one, then we use similar arguments as before to show that $\sum_{(i,j)\in C} \beta^{\hat{q}}_{ij}=0$. Let solution $(x,f) \in UNFM$ use path p for the routing of commodity $\hat{q}$. Next, define solution $(\bar{x},\bar{f})$ to be a solution that employs exactly the same routing for all commodities $q \in Q$, except that commodity $\hat{q}$ is also routed on cycle C. Comparing the two solutions, and using the fact that $\alpha_{ij}=0$ for all $\{i,j\} \in E$, it follows that $\sum_{(i,j)\in C} \beta^{\hat{q}}_{ij}=0$.

If the number of nodes on path p that are also on the cycle C is greater than or equal to 2, then define v_1 as the first, and v_2 to be the last node on the path that is also on the cycle. As a result, path p can be decomposed into three parts p_1, p_2, p_3, where p_1 is a path from $s^{\hat{q}}$ to v_1, p_2 is a path from v_1 to v_2, and p_3 is a path from v_2 to $t^{\hat{q}}$. Similarly, the cycle C can be decomposed into a path C_1 from v_1 to v_2 and a path C_2 from v_2 to v_1. Given these definitions, we can construct two new paths from $s^{\hat{q}}$ to $t^{\hat{q}}$ in the graph. The first path can be represented as p_1, C_1, p_3 and the second path as $p_1, C_2^{\mathrm{r}}, p_3$, where C_2^{r} is the reversed path of C_2. Let $(x,f) \in UNFM$ be a solution that uses the first path for the routing of commodity $\hat{q}$. Given this solution, define a solution $(\bar{x},\bar{f}) \in UNFM$ that employs the same routing strategy for all commodities $q \in Q\setminus\{\hat{q}\}$, but uses the second path for commodity $\hat{q}$. Since both solutions satisfy the equality it follows that $\sum_{(i,j)\in C_1} \beta^{\hat{q}}_{ij} - \sum_{(i,j)\in C_2^{\mathrm{r}}} \beta^{\hat{q}}_{ij}=0$. Exploiting the fact that $\beta^q_{ij}=-\beta^q_{ji}$ for all $q \in Q$ and for all $\{i,j\} \in E$, it follows that $\sum_{(i,j)\in C} \beta^{\hat{q}}_{ij}=\sum_{(i,j)\in C_1} \beta^{\hat{q}}_{ij}+\sum_{(i,j)\in C_2} \beta^{\hat{q}}_{ij}=0$, which proves our intermediate claim.

Next, for all $q \in Q$, for all $i \in V$, and a path p from s^q to i in the graph, let $\mu^q_i=\sum_{(i,j)\in p} \beta^q_{ij}$. We claim that the value of μ^q_i is independent of the selected path p. To verify this claim, let p_1, p_2 be two paths from $s^{\hat{q}}$ to i in the graph, and let $p_1^{\mathrm{r}}, p_2^{\mathrm{r}}$ be the reversed paths. Then $p_1 \cup p_2^{\mathrm{r}}$ forms a cycle, hence, $\sum_{(i,j)\in p_1\cup p_2^{\mathrm{r}}} \beta^q_{ij}=0$. Using $\beta^q_{ij}=-\beta^q_{ji}$ it then follows that $\sum_{(i,j)\in p_1} \beta^q_{ij}=\sum_{(i,j)\in p_2} \beta^q_{ij}$, thus indeed, the value of μ^q_i is independent of the selected path from s^q to i.

If we multiply the flow conservation equalities of the model *UNFM* by these multipliers and add them all up, we obtain the following expression:

$$\begin{aligned}\sum_{q\in Q}\sum_{i\in V}\mu^q_i\left(\sum_j f^q_{ji}-\sum_j f^q_{ij}\right) &= \sum_{q\in Q}\sum_{\{i,j\}\in E}\{(\mu^q_i-\mu^q_j)f^q_{ji}+(\mu^q_j-\mu^q_i)f^q_{ij}\}\\ &= \sum_{q\in Q}\sum_{\{i,j\}\in E}(\beta^q_{ij}f^q_{ij}+\beta^q_{ji}f^q_{ji}).\end{aligned}$$

This implies that the equality is indeed a linear combination of the model equalities. □

For both the path formulations we can prove that the non-negativity constraints $z_p^q \geqslant 0$ are facet defining [12,13]. A similar result holds for the non-negativity constraints of the flow variables $f_{ij}^q \geqslant 0$. Henceforth, we refer to these inequalities as the trivial facets.

4. Network loading problems and the edge capacity polytope

The models presented in the previous section have a lot of similarities. In the flow formulations (1)–(4) and (9)–(13), we can distinguish between the flow conservation constraints (2) (respectively, (10)), and the edge capacity constraints (3) (respectively (11), (12)). In the path formulations (5)–(8) and (14)–(18), the constraints can be divided in path selection (constraints (6), respectively (15)) and edge capacity constraints (respectively (7) and (16), (17)).

In this section we derive relations between the polytopes corresponding to the models of the previous section and polytopes defined by a single (pair of) edge capacity constraint(s). More precisely, we show that non-trivial facet defining inequalities for the polytopes related to a single edge constraint are non-trivial facet defining inequalities for the polytopes corresponding to the original problem.

We define the following sets, which are defined by a single edge capacity constraint (denoted with X) or a pair of edge capacity constraints (denoted with Y).

$$X_{ij}^{\mathrm{UF}} = \left\{ (x,f) \in \mathbb{Z}_0^+ \times \{0,1\}^{2|Q|} : \lambda x_{ij} \geqslant \sum_{q\in Q} d^q (f_{ij}^q + f_{ji}^q) \right\},$$

$$X_{ij}^{\mathrm{BF}} = \left\{ (x,f) \in \mathbb{Z}_0^+ \times \{0,1\}^{|Q|} : \lambda x_{ij} \geqslant \sum_{q\in Q} d^q f_{ij}^q \right\},$$

$$Y_{ij}^{\mathrm{BF}} = \left\{ (x,f) \in \mathbb{Z}_0^+ \times \{0,1\}^{2|Q|} : \lambda x_{ij} \geqslant \sum_{q\in Q} d^q f_{ij}^q,\ \ \lambda x_{ij} \geqslant \sum_{q\in Q} d^q f_{ji}^q \right\},$$

$$X_{ij}^{\mathrm{UP}} = \left\{ (x,z) \in \mathbb{Z}_0^+ \times \{0,1\}^{\sum_{q\in Q} |P_{ij}^q| + |P_{ji}^q|} : \lambda x_{ij} \geqslant \sum_{q\in Q} \sum_{p\in P_{ij}^q \cup P_{ji}^q} d^q z_p^q, \right.$$
$$\left. \sum_{p\in P_{ij}^q \cup P_{ji}^q} z_p^q \leqslant 1,\ \ \forall q \in Q \right\},$$

$$X_{ij}^{\mathrm{BP}} = \left\{ (x,z) \in \mathbb{Z}_0^+ \times \{0,1\}^{\sum_{q\in Q} |P_{ij}^q|} : \lambda x_{ij} \geqslant \sum_{q\in Q} \sum_{p\in P_{ij}^q} d^q z_p^q, \right.$$

$$\left. \sum_{p\in P_{ij}^q} z_p^q \leqslant 1,\ \forall q \in Q \right\},$$

$$Y_{ij}^{\mathrm{BP}} = \left\{ (x,z) \in \mathbb{Z}_0^+ \times \{0,1\}^{\sum_{q\in Q} |P_{ij}^q|+|P_{ji}^q|} : \lambda x_{ij} \geqslant \sum_{q\in Q} \sum_{p\in P_{ij}^q} d^q z_p^q, \right.$$

$$\left. \lambda x_{ij} \geqslant \sum_{q\in Q} \sum_{p\in P_{ji}^q} d^q z_p^q,\ \sum_{p\in P_{ij}^q \cup P_{ji}^q} z_p^q \leqslant 1 \right\}.$$

Obviously any valid inequality for these polytopes is valid for the corresponding original problem. Even stronger, we can prove that for the unidirected models any non-trivial facet defining inequality for these polytopes is also a facet defining inequality for the corresponding original problem. For the bidirected models the same result holds for the edge models that incorporate capacity constraints in both directions on the edge.

Theorem 4. *Any non-trivial facet defining inequality for* $conv(X_{ij}^{\mathrm{UP}})$ *is a non-trivial facet defining inequality for* $conv(UNPM)$.

Proof. Let $a_{ij}x_{ij} \geqslant \sum_{q\in Q} \sum_{p\in P_{ij}^q \cup P_{ji}^q} b_p^q z_p^q - c$ be a non-trivial facet defining inequality for $conv(X_{ij}^{\mathrm{UP}})$. If k denotes the dimension of $conv(X_{ij}^{\mathrm{UP}})$ then $k = 1 + \sum_{q\in Q}(|P_{ij}^q| + |P_{ji}^q|)$, since the polytope $conv(X_{ij}^{\mathrm{UP}})$ is full dimensional. For each $q \in Q$, let $\bar{p}^q \notin P_{ij}^q \cup P_{ji}$ be a path that does not visit arc (i,j) nor (j,i). Now consider the polytope

$$T = conv(\{(x,z) \in UNPM : z_p^q = 0, \forall q \in Q,\ \forall p \notin (P_{ij}^q \cup P_{ji}^q \cup \{\bar{p}^q\})\}).$$

This polytope is the convex hull of the set of solutions for the restricted network loading problem where a commodity $q \in Q$ can only be routed on path $\bar{p}^q$ or on a path that visits edge $\{i,j\}$. Using Proposition 1 the dimension of T thus is $|E| + \sum_{q\in Q}(|P_{ij}^q| + |P_{ji}^q|) = k + |E| - 1$. First we show that the inequality $a_{ij}x_{ij} \geqslant \sum_{q\in Q} \sum_{p\in P_{ij}^q \cup P_{ji}^q} b_p^q z_p^q - c$ is also a facet defining inequality for T by constructing $k + |E| - 1$ affinely independent solution vectors in T that satisfy the inequality at equality. Note that there exist k affinely independent vectors $(\bar{x}, \bar{z}) \in X_{ij}^{\mathrm{UP}}$ that satisfy the inequality at equality. Given such a vector $(\bar{x}, \bar{z})$ we define a vector $(\tilde{x}, \tilde{z}) \in T$ as follows. For all $q \in Q$, let $\tilde{z}_p^q = \bar{z}_p^q$ for all $p \in P_{ij}^q \cup P_{ji}^q$, $\tilde{z}_{\bar{p}}^q = 1$ if $\sum_{p\in P_{ij}^q \cup P_{ji}^q} \bar{z}_p^q = 0$, $\tilde{z}_{\bar{p}}^q = 0$ otherwise, and $\tilde{z}_p^q = 0$ for all $p \notin (P_{ij}^q \cup P_{ji}^q \cup \{\bar{p}^q\})$. Moreover, define $\tilde{x}_{ij} = \bar{x}_{ij}$ and $\tilde{x}_{uv} = \lceil \sum_{q\in Q} d^q \rceil$, for all $\{u,v\} \neq \{i,j\}$. Then these k vectors $(\tilde{x}, \tilde{z}) \in T$ are also affinely independent. Moreover, for any of these given vectors, we can install one additional unit of capacity on any of the edges $\{u,v\} \neq \{i,j\}$, which leads to $|E| - 1$ additional vectors. All of these

$k + |E| - 1$ vectors are affinely independent and satisfy the inequality at equality, hence, the inequality is also facet defining for the polytope T.

Next, we prove that maximal sequential lifting applied to a variable that is fixed to zero in the polytope T yields a lifting coefficient zero, which implies that the inequality is also facet defining for the $conv(UNPM)$. Thus, let $\tilde{q} \in Q$ and let $\tilde{p} \notin (P_{ij}^{\tilde{q}} \cup P_{ji}^{\tilde{q}} \cup \{\bar{p}^{\tilde{q}}\})$. If we apply maximal lifting on the variable $z_{\tilde{p}}^{\tilde{q}}$ to obtain a valid inequality $a_{ij}x_{ij} \geqslant \sum_{q\in Q} \sum_{p\in P_{ij}^q \cup P_{ji}^q} b_p^q z_p^q + b_{\tilde{p}}^{\tilde{q}} z_{\tilde{p}}^{\tilde{q}} - c$, then the lifting coefficient $b_{\tilde{p}}^{\tilde{q}}$ is determined by

$$b_{\tilde{p}}^{\tilde{q}} = \min_{\substack{(x,z)\in UNPM\,:\, z_{\tilde{p}}^{\tilde{q}}=1, \\ z_p^q=0 \ \forall q\in Q \ \forall p\notin (P_{ij}^q \cup P_{ji}^q \cup \{\bar{p}^q, \tilde{p}^{\tilde{q}}\})}} \left\{ a_{ij}x_{ij} - \sum_{q\in Q} \sum_{p\in P_{ij}^q \cup P_{ji}^q} b_p^q z_p^q + c \right\}.$$

Since the facet defining inequality under consideration is non-trivial, there exists a solution $(\tilde{x}, \tilde{z}) \in T$ with $\tilde{z}_{\bar{p}}^{\tilde{q}} = 1$ that satisfies the inequality at equality. Now consider the solution vector that is obtained by replacing path $\bar{p}$ by $\tilde{p}$ for commodity q. This yields a solution vector that is feasible for the minimization lifting problem and has objective value zero since the coefficient of the variable $z_{\bar{p}}^{\tilde{q}}$ is zero in the facet defining inequality. Since the lifting coefficient is non-negative it then follows that it must be zero. Repeating this argument for all remaining variables that are currently fixed to zero yields the desired result. □

Theorem 5. *Any non-trivial facet defining inequality for $conv(Y_{ij}^{\mathrm{BP}})$ is a non-trivial facet defining inequality for $conv(BNPM)$.*

Proof. Similar to the proof of Theorem 4. □

Theorem 6. *Any non-trivial facet defining inequality for $conv(X_{ij}^{\mathrm{UF}})$ is a non-trivial facet defining inequality for $conv(UNFM)$.*

Proof. Analogous to the proof of Theorem 4. □

Theorem 7. *Any non-trivial facet defining inequality for $conv(Y_{ij}^{\mathrm{BF}})$ is a non-trivial facet defining inequality for $conv(BNFM)$.*

Proof. Analogous to the proof of Theorem 5. □

The relation between the polytopes $conv(X_{ij}^{\mathrm{BP}})$ and $conv(BNPM)$ (as well as between $conv(X_{ij}^{\mathrm{BF}})$ and $conv(BNFM)$) is more complicated. Theorems 4–7 are proved by considering a projection of the network loading polytope. First, it is proved that the inequality for the single edge polytope defines a facet of this projection. Next, it is proved that the maximum lifting coefficient of all projected variables equals zero. The maximum lifting coefficient of the variables projected out for the relation between $conv(X_{ij}^{\mathrm{BP}})$ and $conv(BNPM)$ is not necessarily zero. Additional conditions have to be satisfied in order to obtain lifting coefficients zero. In Proposition 8, we give three equivalent characterizations of the conditions under which a non-trivial facet defining inequality for the single edge polytope defines a facet defining inequality for the network loading polytope.

Proposition 8. *The following statements are equivalent*:

(i) *An inequality* $ax \geqslant b^T z - c$ *which is a non-trivial facet defining inequality for* $conv(X_{ij}^{\mathrm{BP}})$ *is a non-trivial facet defining inequality for* $conv(BNPM)$.
(ii) $\forall \hat{q} \in Q_{ji} : \exists \bar{Q} \subseteq Q_{ij} \setminus \{\hat{q}\}, \sum_{q\in\bar{Q}} b^q - c = a \max\{\lceil d^{\hat{q}} \rceil, \lceil \sum_{q\in\bar{Q}} d^q \rceil\}$.
(iii) $\forall \hat{q} \in Q_{ji} : \exists \bar{Q} \subseteq Q_{ij} \setminus \{\hat{q}\}, \lceil \sum_{q\in\bar{Q}} d^q \rceil \geqslant \lceil d^{\hat{q}} \rceil$ *and* $\sum_{q\in\bar{Q}} b^q - c = a\lceil \sum_{q\in\bar{Q}} d^q \rceil$.
(iv) $\forall \hat{q} \in Q_{ji}$: *the maximization problem*

$$\begin{aligned} \max \quad & \theta \\ \text{s.t.} \quad & a\theta = \sum_{q\in Q_{ij}\setminus\{\hat{q}\}} b^q w^q - c, \\ & \theta \geqslant \sum_{q\in Q_{ij}\setminus\{\hat{q}\}} d^q w^q, \\ & \theta \in \mathbb{Z}_0^+, \; w^q \in \{0,1\}, \quad \forall q \in Q_{ij} \setminus \{\hat{q}\} \end{aligned}$$

has an optimal objective value $\theta^* \geqslant \lceil d^{\hat{q}} \rceil$.

Proof. It is fairly easy to see that (ii) $\Rightarrow$ (iv) $\Rightarrow$ (iii) $\Rightarrow$ (ii). Hence, we will restrict ourselves to prove that (i) $\Leftrightarrow$ (ii). Suppose (ii) holds. Similar as in the proof of Theorem 5 we can lift variables $z_{\hat{p}}^{\hat{q}}$ for $\hat{q} \in Q_{ji}$ and $\hat{p} \in P_{ji}^{\hat{q}}$. The maximal lifting coefficient $b_{\hat{p}}^{\hat{q}}$ for such a variable equals

$$b_{\hat{p}}^{\hat{q}} = \min_{z_{\hat{p}}^{\hat{q}}=1} \left\{ ax_{ij} - \left(\sum_{q\in Q_{ij}\setminus\{\hat{q}\}} \sum_{p\in P_{ij}^q} b_p^q z_p^q - c \right) \right\}.$$

It is easy to see that $b_{\hat{p}}^{\hat{q}} \geqslant 0$ since otherwise the starting inequality was not valid. The conditions of (ii) now give that the minimum is indeed zero. This argument can be repeated for all variables $z_{\hat{p}}^{\hat{q}}$ for $\hat{q} \in Q_{ji}$ and $\hat{p} \in P_{ji}^{\hat{q}}$. As a consequence, the inequality is facet defining for $conv(Y_{ij}^{\mathrm{BP}})$. Now, Theorem 5 gives the desired result.

The reversed claim is easy to see. If no subset $\bar{Q} \subseteq Q_{ij} \setminus \{\hat{q}\}$ satisfies the conditions as posed in (ii), then the lifting coefficient as determined by the minimization problem described in the above, will not be equal to zero. Hence, the inequality can be strengthened, and does not define a facet of $conv(BNPM)$. $\square$

Proposition 9. *The following statements are equivalent*:

(i) *An inequality* $ax \geqslant b^T f - c$ *which is a non-trivial facet defining inequality for* $conv(X_{ij}^{\mathrm{BF}})$ *is a non-trivial facet defining inequality for* $conv(BNFM)$.
(ii) $\forall \hat{q} \in Q_{ji} : \exists \bar{Q} \subseteq Q_{ij} \setminus \{\hat{q}\}, \sum_{q\in\bar{Q}} b^q - c = a \max\{\lceil d^{\hat{q}} \rceil, \lceil \sum_{q\in\bar{Q}} d^q \rceil\}$.
(iii) $\forall \hat{q} \in Q_{ji} : \exists \bar{Q} \subseteq Q_{ij} \setminus \{\hat{q}\}, \lceil \sum_{q\in\bar{Q}} d^q \rceil \geqslant \lceil d^{\hat{q}} \rceil$ *and* $\sum_{q\in\bar{Q}} b^q - c = a\lceil \sum_{q\in\bar{Q}} d^q \rceil$.

(iv) $\forall \hat{q} \in Q_{ji}$: *the maximization problem*

$$\begin{aligned} \max \quad & \theta \\ \text{s.t.} \quad & a\theta = \sum_{q \in Q_{ij} \setminus \{\hat{q}\}} b^q w^q - c, \\ & \theta \geqslant \sum_{q \in Q_{ij} \setminus \{\hat{q}\}} d^q w^q, \\ & \theta \in \mathbb{Z}_0^+, \ w^q \in \{0,1\}, \quad \forall q \in Q_{ij} \setminus \{\hat{q}\}, \end{aligned}$$

has an optimal objective value $\theta^* \geqslant \lceil d^{\hat{q}} \rceil$.

Proof. Analogous to the proof of Proposition 8. □

5. Valid inequalities

In this section, we present several classes of valid inequalities that are known for the edge capacity polytope and the network loading polytope in general.

5.1. Valid inequalities for the edge capacity polytope

In Section 4 we introduced six different polytopes restricted to a single edge of the original model. The edge models X_{ij}^{UF}, X_{ij}^{UP}, X_{ij}^{BF}, X_{ij}^{BP} are similar. They describe a knapsack with variable integer capacity. Since the associated polyhedra are the same we use easier notation and a redefinition of the edge capacity model that captures all of the aforementioned edge models. Consider a set Q of items (commodities) and let $d^q \in \mathbb{Q}^+$ represent the size (demand) for an item $q \in Q$ (normalized to the base capacity λ). Let the integer variable x denote the number of capacity units selected and let the binary variables f^q indicate whether or not an individual item q is selected. The edge capacity set is then defined as

$$X = \left\{ (x, f) \in \mathbb{Z}_0^+ \times \{0,1\}^{|Q|} : x \geqslant \sum_{q \in Q} d^q f^q \right\}.$$

The problem defined by X and an arbitrary objective function is $\mathcal{NP}$-hard [13]. The edge capacity polytope was first studied by Brockmüller et al. [8,9] in the context of the unidirected network loading problem. They derived the class of *c-strong inequalities*. These inequalities were generalized by van de Leensel [13] (see also [12]), and independently by Atamturk and Rajan [1]. Both groups derived the class of *lifted knapsack covers*. For a fixed x, the polytope $conv(X)$ reduces to the knapsack polytope. Therefore, valid inequalities for the knapsack polytope can be lifted to valid inequalities for the edge capacity polytope X. In particular, for minimal knapsack covers [2,16,19] the lifting can be done in polynomial time and results in one or two facet defining inequalities. Let Q^0, Q^1, S be a partition of Q, and let

$$X(Q^0, Q^1) = \{(x, f) \in X : f^q = 0 \ \forall q \in Q^0, \ \ f^q = 1 \ \forall q \in Q^1\}.$$

If S defines a minimal cover for the knapsack defined by $X(Q^0,Q^1)$ and $x=\bar{x}$, i.e., $\sum_{q\in Q^1\cup S} d^q > \bar{x}$ and $\sum_{q\in Q^1\cup S\setminus\{i\}} d^q \leqslant \bar{x}$ for all $i\in S$. Then

$$x \geqslant \sum_{q\in S} D^q f^q + D(Q^1\cup S) - \sum_{q\in S} D^q,$$

$$\alpha^{\mathrm{U}} x \geqslant \sum_{q\in S} \{\alpha^{\mathrm{U}}(D^q-1)+1\} f^q + \alpha^{\mathrm{U}}\left(D(Q^1\cup S) - \sum_{q\in S} D^q\right) + (\alpha^{\mathrm{U}}-1)(|S|-1),$$

define two facet defining inequalities for $X(Q^0,Q^1)$, with

$$\alpha^{\mathrm{U}} = \min_{k=0,1,\dots,|S|-2} \frac{|S|-1-k}{D(Q^1\cup S)-1-D(Q^1\cup S_k)} \geqslant 1,$$

the maximum lifting coefficient of integer lifting of variable x. Here, $D^q = \lceil d^q \rceil$ is the smallest integer larger than d^q, whereas $D(S) = \lceil \sum_{q\in S} d^q \rceil$ denotes the same for a subset $S\subseteq Q$ of the commodities. The set S_k contains the k first elements of S, sorted by the fractional part of their demand $d^q - \lfloor d^q \rfloor$ in non-decreasing order. Inequality (19) defines a different facet, i.e., $\alpha^{\mathrm{U}} > 1$, if and only if $D(S_{|S|-1}) = D(S_{|S|-2})$. Lifting of the projected variables f^q, $q\in Q^0\cup Q^1$, can be done in $\mathcal{O}(n^3)$ [12].

The class of lifted knapsack covers is contained in the larger class of *lower convex envelope inequalities*. However, only for the subclass of lifted knapsack covers, it can be proved that the inequalities define facets for *conv*(X). For details on lower convex envelope inequalities as well as implementation issues we refer to [12,13].

Similar to X, we can define a common set Y for the models Y_{ij}^{BF} and Y_{ij}^{BP}.

$$Y = \left\{(x,f,h)\in \mathbb{Z}_0^+ \times \{0,1\}^{2|Q|} : x \geqslant \sum_{q\in Q} d^q f^q, x \geqslant \sum_{q\in Q} d^q h^q \right\}.$$

For the bidirected edge capacity polytope *conv*(Y), in [13] the class of *two-side inequalities* is derived for X. Let $\hat{q}\in Q$ and let $\alpha\in\mathbb{Z}^+$ such that $1\leqslant\alpha\leqslant D^{\hat{q}}$. Then

$$x \geqslant \alpha f^{\hat{q}} + \sum_{q\in Q} (D^q-\alpha) h^q \tag{19}$$

is a valid inequality for Y. Moreover, one can specify conditions for the special case, where $\alpha=1$ such that the corresponding inequality is facet defining for *conv*(Y) (see [12,13]).

5.2. Valid inequalities for the network loading polytope

Apart from the inequalities for the edge capacity polytope, several other classes of valid inequalities are known for network loading problems. In the computational study presented in the next section, we incorporated two of these classes. Cut-set inequalities are used quite extensively for network loading problems (see for instance Barahona [3], Magnanti et al. [14,15], Bienstock and Günlük [7], among others). Given a partition of the node set V into two sets S and T, let $d[S,T]$ denote the accumulated demand of all commodities with source node in S and sink node in T. Then it is clear that the total capacity on the edges in the cut $\delta[S,T]$ should exceed this accumulated demand since all of these commodities must cross the cut. Since, capacity can only be installed in integer amounts, the cut-set inequalities read

$$\sum_{\{i,j\}\in\delta[S,T]} x_{ij} \geqslant \lceil \max\{d[S,T], d[T,S]\}\rceil$$

for the bidirected capacity models *BNFM* and *BNPM*, and

$$\sum_{\{i,j\}\in\delta[S,T]} x_{ij} \geqslant \lceil d[S,T] + d[T,S]\rceil$$

for the unidirected versions *UNFM* and *UNPM*. Likewise, three partition inequalities (based on a partition of the node set into three sets) have been considered (see [7]), as well as the general K-cuts (see [4]). Bienstock [5] proved that separation of cut-set inequalities is $\mathcal{NP}$-hard even if the input is restricted to a vector that satisfies all model equations (see [8]). Our instances, however, only have a limited number of vertices, which made it possible to enumerate all cut-set inequalities for $K = 2$ and 3 in a separation routine.

The cut-set inequalities are facet defining for bifurcated network loading. For non-bifurcated routing, however, the inequalities can be strengthened. Computational experiments showed that in practice it is ineffective to apply the time-consuming lifting procedure within a branch-and-cut algorithm.

6. Computational results

In this section, we compare the performance of the inequalities of the previous section within cutting plane algorithms for the diverse models of Section 2. The aim of our comparison is twofold. On the one hand, we would like to compare the solvability of the models with and without the separation of inequalities related to the edge capacity polytope. On the other hand, we would like to compare the solvability of the unidirected and bidirected model. For these purposes, we implemented a branch-and-cut algorithm with help of the C++ framework a branch-and-cut system (ABACUS), version 2.2 [17]. ABACUS uses CPLEX 6.5 [10] as linear programming solver. For stand-alone integer linear programs, CPLEX 7.1 is used.

To compare the effectiveness of the derived inequalities for the unidirected and bidirected problems, we tested two different versions (for the different models) of our branch-and-cut algorithm on the same set of instances, that is provided by KPN Research in Leidschendam, The Netherlands. These instances are originally generated for bidirected capacity installation in ATM networks, but can also be used to test the algorithm for the unidirected capacity version. Our comparison focuses on the flow formulations *UNFM* and *BNFM*. Computational experiments indicated that for larger graphs the exponential growth of the number of path variables results in a decrease of the performance of the path formulations in comparison with the flow formulations.

Table 1
Computational results branch-and-cut for *UNFM*

Instance	z_{LP}	z_{LP+}	z_{IP}	z_{UB}	No. nodes	No. cuts	CPU time (s)
kpn_4_3	3.10	4.75	5	5	1	17	0.11
kpn_4_10	10.32	11.90	12	12	1	16	0.08
kpn_4_20	20.64	22.60	23	23	1	14	0.08
kpn_5_3	3.37	5.33	6	6	1	42	0.21
kpn_5_10	11.23	13.28	14	14	1	50	0.35
kpn_5_20	22.45	25.25	26	26	1	76	0.82
kpn_6_3	3.37	5.56	7	7	15	501	16.40
kpn_6_10	11.23	13.12	14	14	1	88	0.63
kpn_6_20	22.45	24.54	27*	27	58547*	111*	1865.47*
kpn_7_3	3.39	5.99	7	7	3	278	4.03
kpn_7_10	11.29	13.46	15	15	47	758	52.62
kpn_7_20	22.58	24.88	27*	27	29870*	208*	3290.39*
kpn_8_3	3.68	6.51	8	8	19	1920	122.58
kpn_8_10	12.26	14.78	17*	19	756671*	524*	47858.17*
kpn_8_20	24.52	26.92	$\geqslant$ 29*	31	> 600000*	596*	> 174000

*These results are obtained by the described two-step procedure.

In Table 1, the results of the branch-and-cut algorithm for the unidirected model *UNFM* are presented, whereas in Table 2, the results for the bidirected case *BNFM* are summarized. The 15 instances are defined on complete graphs with 4 to 8 nodes. For each graph size, three instances were generated by multiplying the (actual) demand by 3, 10, and 20. The name of each instance, stated in the first column, refers to the number of nodes in the graph (first digit), and the demand multiplication factor (second digit). For all instances it holds that the installation costs equal one for all edges. The next four columns report on the value of the linear relaxation (z_{LP}), the linear relaxation plus violated inequalities in the root node (z_{LP+}), the optimal value (z_{IP}), and an upper bound used to limit the size of the branch-and-cut tree (z_{UB}). The upper bounds were computed with the heuristics available in the software package UMBRIA, developed by order of KPN Research [13]. The last three columns summarize statistics concerning the number of nodes of the branch-and-cut tree (# nodes), the number of cutting planes added (# cuts), and the total CPU time in seconds. The computations are performed on a Sun Ultra-1 170 E Creator workstation with 512MB internal memory.

The results of Table 1 show that the smaller instances can be solved in the root node, because the lower bound provided by cutting planes is within one of the upper bound. However, as soon as the gap between lower and upper bound becomes larger, the effort increases to find the optimal solution. As long as the gap in the root is bounded by two, the branch-and-cut algorithm succeeds to find the optimum. However, mainly due to the excessive memory requirements of ABACUS, the algorithm runs out of memory for the instances with gap larger than two. For those cases, we apply a

Table 2
Computational results branch-and-cut for *BNFM*

Instance	z_{LP}	z_{LP+}	z_{IP}	z_{UB}	No. nodes	No. cuts	CPU time (s)
kpn_4_3	1.74	3.00	3	3	1	34	0.13
kpn_4_10	5.81	7.19	8	8	1	35	0.15
kpn_4_20	11.61	13.00	13	13	1	26	0.13
kpn_5_3	1.89	3.75	4	4	1	74	0.34
kpn_5_10	6.32	8.50	9	9	1	65	0.36
kpn_5_20	12.64	15.06	16	16	1	156	0.90
kpn_6_3	1.94	4.11	5	5	1	147	1.14
kpn_6_10	6.45	8.36	9	10	33	1,891	33.71
kpn_6_20	12.90	15.47	16	17	125	3,831	89.45
kpn_7_3	1.95	4.42	6	6	19	1,628	65.90
kpn_7_10	6.52	8.81	10	10	19	1,568	64.71
kpn_7_20	13.03	15.73	17	18	255	7,093	415.06
kpn_8_3	2.17	4.89	7*	7	46280*	1297*	3353.99*
kpn_8_10	7.22	9.83	11*	12	36346*	950*	3641.29*
kpn_8_20	14.45	17.33	19*	21	551837*	754*	163613.36*

*These results are obtained by the described two-step procedure.

two-step procedure. First, we strengthen the linear relaxation in the root node with valid inequalities, until the separation procedures implemented in ABACUS do not lead to new inequalities anymore. Then, the resulting linear program is solved directly with CPLEXs branch-and-bound algorithm. The time is the sum of both steps, whereas the number of nodes results from CPLEX, and the number of cuts from ABACUS. In this way, the instances kpn_6_20, kpn_7_20, and kpn_8_10 could be solved to optimality. However, the number of nodes needed by CPLEX is enormous. For this reason, also this two-step procedure was not able to solve instance kpn_8_20.

Experiments show that the hardness of the instances is mainly due to the non-bifurcated routing restriction. Without this restriction (i.e., bifurcated routing), all problems could be solved within reasonable time. As the optimal solution of bifurcated routing provides a lower bound for non-bifurcated routing, for several (small) instances optimality could be proved with this bound.

The results for the bidirected case in Table 2 have many similarities with those of the unidirected case. Again, for small instances, the gap between heuristic and linear programming solution can be closed in the root node almost completely, resulting in a proof of optimality in many cases. Due to a non-optimal heuristic solution for the instances kpn_6_10 and kpn_6_20, these instances cannot be solved in the root node, although the gap is within one of optimal. For the remaining instances, ABACUS can solve the problems with 7 nodes, but not with 8 nodes due to the huge memory requirements. The two-step procedure described for the unidirected case results also in this case to optimal solutions.

Table 3
Solvability of *UNFM* with and without cutting planes in root of B&B tree

	Statistics				Gap closed by	CPU time (s)		No. nodes		No. cuts
Instance	$\|V\|$	$\|E\|$	$\|Q\|$	λ		B&B	C&B	B&B	C&B	C&B
unfm08a	8	12	52	8	64.66	3.61	6.40	128	91	43
unfm08b	8	12	52	16	76.52	13.80	8.12	332	117	70
unfm08c	8	12	52	32	92.82	4.34	7.65	134	217	139
unfm10a	10	14	64	8	76.72	58.60	30.20	2796	418	36
unfm10b	10	14	64	16	74.87	38.70	4.05	627	7	87
unfm10c	10	14	64	32	60.30	25.70	84.00	245	1200	109
unfm15a	15	22	148	8	75.62	5.13e+4	5.47e+4	220807	115587	87
unfm15b	15	22	148	16	93.83	6.66e+3	9.97e+3	17132	19373	170
unfm15c	15	22	148	32	100.00	818.00	746.00	2276	1418	332
unfm17a	17	22	222	8	87.60	>3.00e+5	707.00	>750000	239	66
unfm17b	17	22	222	16	>83.89	>4.18e+5	>7.63e+5	>614500	>514000	135
unfm17c	17	22	222	32	98.57	4.43e+4	2.79e+3	46274	2177	325

Let us now compare the results of *UNFM* with these of *BNFM*. The tables show that the solvability of the *UNFM* model depends more heavily on the size of the demands as it is for the *BNFM* model. The branch-and-cut algorithm cannot solve the largest instances (in demand) with 6 and 7 nodes for the *UNFM* model, whereas it can solve the same instances for the *BNFM*. On the other hand, the smallest instance with 8 nodes can be solved by the *UNFM* model, where it cannot be solved for the *BNFM* model. The percentage with which the gap is closed on average is almost equal for both models (at least 66.9% for *UNFM* against 68.5% for *BNFM*). Nevertheless, for both models and instances as small as 8 nodes (and all edges), finding an optimal solution (and proving it) seems to be too hard for state-of-the-art branch-and-cut algorithms.

For non-complete graphs, the cutting planes can help to solve larger network loading problems to optimality more efficiently. To show the effect of the inequalities on the solvability, we compare two strategies on a second set of instances. These instances originate from a study to the design of optical telecommunication networks at ZIB. We compare the effectiveness of the mixed integer programming solver of CPLEX (version 7.1) with and without the addition of cutting planes in the root node; strategies branch-and-bound (B&B) and cut-and-branch (C&B). Tables 3 and 4 show, respectively, the results for both strategies for the *UNFM* and *BNFM* model. The column *gap closed by* denotes the percentage with which the gap between LP relaxation and optimal value is by the cutting planes added in the root node of the cut-and-branch strategy. The computations are performed on a PC with a Pentium III 800 MHz processor, 512 MB internal memory, and Linux as operating system. From these tables, we can conclude that the effectiveness of the strategies relies heavily on the particular instance to be solved. Instance `unfm15a` seems to be extremely difficult, even with the addition of cutting planes in the root node, whereas instance `unfm15c` is relatively

Table 4
Solvability of *BNFM* with and without cutting planes in root of B&B tree

	Statistics				Gap closed by	CPU time (s)		No. nodes		No. cuts
Instance	$\|V\|$	$\|E\|$	$\|Q\|$	λ		B&B	C&B	B&B	C&B	C&B
bnfm08a	8	12	52	8	82.69	125.0	5.3	18136	4	157
bnfm08b	8	12	52	16	62.74	16.1	21.8	674	188	334
bnfm08c	8	12	52	32	78.33	18.2	12.9	846	39	382
bnfm10a	10	14	64	8	79.86	369.0	49.1	17815	242	264
bnfm10b	10	14	64	16	86.59	16.1	37.2	215	32	440
bnfm10c	10	14	64	32	75.01	22.1	51.2	224	208	333
bnfm15a	15	22	148	8	82.86	>9.0e+5	4.91e+4	>1470000	56780	448
bnfm15b	15	22	148	16	>76.04	>5.0e+5	>4.25e+5	> 735000	>461000	753
bnfm15c	15	22	148	32	78.85	6.68e+3	1.11e+3	11075	1290	1420
bnfm17a	17	22	222	8	81.98	>4.7e+5	6.10e+3	> 967000	4400	327
bnfm17b	17	22	222	16	88.09	5.45e+4	2.36e+3	64205	1873	6176
bnfm17c	17	22	222	32	76.37	1.73e+4	>2.39e+5	21878	>526000	1468

simple. Remarkable is that the addition of cutting planes has not always a positive effect on the performance of the integer programming solver. For example, the instance `unfm10c` can be solved more than three times faster without cutting planes than with them. On the other hand, the C&B strategy clearly outperforms B&B on several instances like `unfm10b`, `unfm17a`, or `bnfm10a`.

Concluding, the success of the cut-and-branch approach on the overall problem depends on the actual instance at hand. Sometimes, the benefits of the polyhedral knowledge are very limited, and sometimes, the same knowledge is the key to solving the problem.

References

[1] A. Atamturk, D. Rajan, On splittable and unsplittable flow capacitated network design arc-set polyhedra, Math. Programming Ser. A 92 (2) (2002) 315–333.
[2] E. Balas, Facets of the knapsack polytope, Math. Programming 8 (1975) 146–164.
[3] F. Barahona, Network design using cut inequalities, SIAM J. Optim. 6 (1996) 823–837.
[4] F. Barahona, On the k-cut problem, Oper. Res. Lett. 26 (3) (2000) 99–105.
[5] D. Bienstock, private communication, 1998.
[6] D. Bienstock, S. Chopra, O. Günlük, C.-Y. Tsai, Minimum cost capacity installation for multicommodity network flows, Math. Programming 81 (1998) 177–199, working paper.
[7] D. Bienstock, O. Günlük, Capacitated network design—polyhedral structure and computation, ORSA J. Comput. 8 (1996) 243–260.
[8] B. Brockmüller, O. Günlük, L.A. Wolsey, Designing private line networks—polyhedral analysis and computation, Discussion Paper 9647, Center for Operations Research and Econometrics, October 1996.
[9] B. Brockmüller, O. Günlük, L.A. Wolsey, Designing private line networks—polyhedral analysis and computation, Discussion Paper 9647 revised, Center for Operations Research and Econometrics, March 1998.

[10] CPLEX division of ILOG. CPLEX callable library, version 6.5, 1999.
[11] B. Gavish, K. Altinkemer, Backbone network design tools with economic tradeoffs, ORSA J. Comput. 2 (3) (1990) 236–252.
[12] C.P.M. van Hoesel, A.M.C.A. Koster, R.L.M.J. van de Leensel, M.W.P. Savelsbergh, Polyhedral results for the edge capacity polytope, Technical Report ZIB 00-22, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Berlin, Germany, 2000. Available at http://www.zib.de/koster/.
[13] R.L.M.J. van de Leensel, Models and algorithms for telecommunication network design, Ph.D. Thesis, Maastricht University, 1999.
[14] T.L. Magnanti, P. Mirchandani, R. Vachani, The convex hull of two-core capacitated network design problems, Math. Programming 60 (1993) 223–250.
[15] T.L. Magnanti, P. Mirchandani, R. Vachani, Modeling and solving the two-facility capacitated network loading problem, Oper. Res. 43 (1995) 142–157.
[16] M. Padberg, A note on zero-one programming, Oper. Res. 23 (1975) 833–837.
[17] S. Thienel, ABACUS: A Branch And CUt System, version 2.2, 1998.
[18] R. Wessäly, Dimensioning Survivable Capacitated NETworks, Ph.D. Thesis, Technische Universität Berlin, 2000.
[19] L.A. Wolsey, Faces of linear inequalities in 0–1 variables, Math. Programming 8 (1975) 165–178.