Capacitated network design using general flow-cutset inequalities

Christian Raack, *Zuse Institute Berlin (ZIB), Takustr. 7, D-14195 Berlin, raack@zib.de*
Arie M.C.A. Koster,* *Zuse Institute Berlin (ZIB), Takustr. 7, D-14195 Berlin, koster@zib.de*
Sebastian Orlowski, *Zuse Institute Berlin (ZIB), Takustr. 7, D-14195 Berlin, orlowski@zib.de*
Roland Wessäly, *Zuse Institute Berlin (ZIB), Takustr. 7, D-14195 Berlin, wessaely@zib.de*

Abstract

This paper deals with directed, bidirected, and undirected capacitated network design problems. We generalize flow-cutset inequalities to these three link types and to an arbitrary modular link capacity structure, and propose a generic separation algorithm. In an extensive computational study on 54 instances from the Survivable Network Design Library (SNDlib), we show that the performance of CPLEX can significantly be enhanced by this class of cutting planes.

Keywords: Capacitated network design, mixed integer rounding, flow-cutset inequalities, SNDlib

Introduction

Capacitated network design is the task to assign capacity to the links of a potential network topology by selecting capacity modules (wavelengths, leased lines, STM-N capacities) from a discrete set such that given communication demands can be satisfied and total installation cost is minimized. In this paper we revisit polyhedral approaches based on cutsets for directed, bidirected and undirected link models [1, 4, 5, 7, 11]. Using mixed integer rounding (MIR), we generalize flow-cutset inequalities to these three link types and to an arbitrary modular link capacity structure and propose a generic separation algorithm.

Our computational study comprises 54 instances covering all SNDlib [17] networks, the link types DIRECTED, UNDIRECTED, and BIDIRECTED as well as the capacity models MODULAR and EXPLICIT defined in SNDlib. In all 54 cases, separating general flow-cutset inequalities improved the performance of CPLEX. In particular, 27 instances could be solved to optimality within one hour of computation time, compared to 17 with the default settings of CPLEX. The best solutions are available at `http://sndlib.zib.de`.

A network instance is given by a directed graph $G = (V, A)$ (DIRECTED link model) or an undirected graph $G = (V, E)$ (BIDIRECTED and UNDIRECTED link model), a set M of installable modules, and a set K of commodities. For undirected graphs $G = (V, E)$ we define $A := \{(i, j), (j, i) : e = ij \in E\}$ to be the set of arcs obtained by bidirecting all edges of E. A module $m \in M$ has a capacity $c^m \in \mathbb{Z}_+ \backslash \{0\}$. With every $k \in K$ we associate a vector $d^k \in \mathbb{Z}^V$ of demands such that $\sum_{v \in V} d^k_v = 0$, assuming a fractional multi-commodity flow routing.

Let $\mathcal{N} \in \{0, 1, -1\}^{V \times A}$ be the node-arc incidence matrix corresponding to V and A and define x^m_a, x^m_e to be the number of installed modules of type $m \in M$ on arc $a \in A$ or edge $e \in E$, respectively. Furthermore, let $f^k \in \mathbb{R}^A_+$ be the flow vector corresponding to commodity $k \in K$. The network design polyhedra for the link models DIRECTED, BIDIRECTED, and UNDIRECTED are given by $P_{di} := \mathrm{conv}(X_{di})$, $P_{bi} := \mathrm{conv}(X_{bi})$, and $P_{un} := \mathrm{conv}(X_{un})$, where

$$
\begin{aligned}
X_{di} &= \{(f, x) \in \mathbb{R}^{A \times K}_+ \times \mathbb{Z}^{A \times M}_+ : \mathcal{N} f^k = d^k \ \forall k \in K, \ \sum_{k \in K} f^k_a \le \sum_{m \in M} c^m x^m_a \ \forall a \in A\}, \\
X_{bi} &= \{(f, x) \in \mathbb{R}^{A \times K}_+ \times \mathbb{Z}^{E \times M}_+ : \mathcal{N} f^k = d^k \ \forall k \in K, \ \max(\sum_{k \in K} f^k_{ij}, \sum_{k \in K} f^k_{ji}) \le \sum_{m \in M} c^m x^m_e \ \forall e{=}ij \in E\}, \\
X_{un} &= \{(f, x) \in \mathbb{R}^{A \times K}_+ \times \mathbb{Z}^{E \times M}_+ : \mathcal{N} f^k = d^k \ \forall k \in K, \ \sum_{k \in K} (f^k_{ij} + f^k_{ji}) \le \sum_{m \in M} c^m x^m_e \ \forall e{=}ij \in E\}.
\end{aligned}
$$

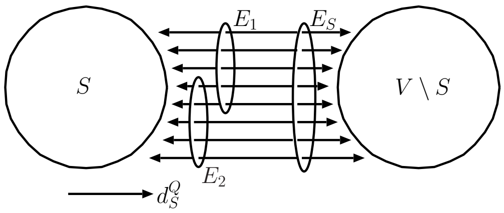
In addition to this MODULAR capacity model, we also consider the EXPLICIT capacity model where the constraints $\sum_{m \in M} x^m_a \le 1$, $a \in A$, are added to the DIRECTED formulation (similarly for the other link models). The objective is to minimize a linear function incorporating flow and module cost over these polyhedra.

*Since 1st April 2007: University of Warwick, Centre for Discrete Mathematics and its Applications (DIMAP), Coventry CV4 7AL, United Kingdom

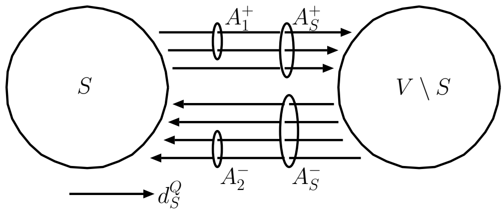
General flow-cutset inequalities

In this section, we present flow-cutset inequalities generalizing those which have been presented in [1, 5, 7, 11, 12, 13] for special cases of the network design problems considered in this paper. We derive them by aggregating model constraints on a cut in the network and applying a subadditive MIR function to the coefficients of the resulting inequality. As shown in [1, 16], these inequalities are facet-defining for the network design polyhedra P_{di}, P_{un}, and P_{bi} under rather general conditions.

We will now introduce the necessary notation related to the network cut, followed by the definition of the subadditive MIR function. Consider a cut defined by a subset S of the supply nodes V and let Q be a subset of the commodities K. For UNDIRECTED and BIDIRECTED models, let E_S be the undirected cut $\{e = ij \in E : i \in S,\, j \in V \backslash S\}$ with (not necessarily disjoint) subsets $E_1, E_2 \subseteq E_S$ (see Figure 1(a)). Similarly, for DIRECTED models, define $A_S^+ := \{(i,j) \in A : i \in S,\, j \in V \backslash S\}$ and $A_S^- := \{(i,j) \in A : j \in S,\, i \in V \backslash S\}$, and consider subsets $A_1^+ \subseteq A_S^+$, $A_2^- \subseteq A_S^-$, and $\bar{A}_1^+ := A_S^+ \backslash A_1^+$ (see Figure 1(b)). When speaking of flow across the cut, the sets A_1^+, $\bar{A}_1^+$, und A_2^- are also used for UNDIRECTED and BIDIRECTED models. In this case, the sets A_1^+, $\bar{A}_1^+$ correspond to forward flow on edges of E_1 and its complement with respect to the cut while A_2^- corresponds to backward flow on edges of E_2.



(a) Undirected cut and selected cut edges



(b) Directed cut and selected cut arcs

Figure 1: Network cuts

Let $f^Q(A^*)$ denote the total flow on some subset A^* of the arcs A with respect to Q, i. e., $f^Q(A^*) := \sum_{k \in Q} \sum_{a \in A^*} f_a^k$, and let $x^m(A^*) := \sum_{a \in A^*} x_a^m$ be the total number of modules of type $m \in M$ on arcs of A^*. The value $x^m(E^*) := \sum_{e \in E^*} x_e^m$ is defined analogously for undirected edges $E^* \subseteq E$.

In the following, the nodeset S and the commodity subset Q are fixed. Let $d_S^Q := \sum_{v \in S} \sum_{k \in Q} d_v^k$ be the total demand value with respect to Q over the cut defined by$S$, where $d_S^k := d_S^{\{k\}}$ for $k \in Q$. Furthermore, let $K_S^+ := \{k \in K : d_S^k > 0\}$ and $K_S^- := \{k \in K : d_S^k < 0\}$ be the commodity subsets with positive and negative demand over the cut, respectively. Without loss of generality we can assume that $K_S^- = \emptyset$ for UNDIRECTED link models by reversing demand directions (which is done implicitly in our algorithms).

We will now define the subadditive function used to derive general flow-cutset inequalities in Proposition 1. Let $a, c, d \in \mathbb{R}$ with $c > 0$ and $\frac{d}{c} \notin \mathbb{Z}$, and define $a^+ := \max(0, a)$ and $a^- := \min(0, a)$. Furthermore, let $r(a,c) := a - c(\lceil \frac{a}{c} \rceil - 1) > 0$ be the remainder of the division of a by c if $\frac{a}{c} \notin \mathbb{Z}$, and c otherwise. A function $F : \mathbb{R} \to \mathbb{R}$ is called *subadditive* if $F(a) + F(b) \geq F(a+b)$ and *superadditive* if $F(a) + F(b) \leq F(a+b)$ for all $a, b \in \mathbb{R}$.

Lemma 1. *The function $F_{d,c} : \mathbb{R} \to \mathbb{R}$ defined by*

$$F_{d,c}(a) := \lceil \tfrac{a}{c} \rceil r(d,c) - (r(d,c) - r(a,c))^+$$

is subadditive and nondecreasing with $F_{d,c}(0) = 0$ and $\bar{F}_{d,c}(a) := \lim_{t \searrow 0} \frac{F_{d,c}(at)}{t} = a^+$ for all $a \in \mathbb{R}$.

Proof. Define $f_x := x - \lfloor x \rfloor$ for $x \in \mathbb{R}$ and let $\varphi_d : \mathbb{R} \to \mathbb{R}$ defined by

$$\varphi_d(a) := \lfloor a \rfloor + \frac{(f_a - f_d)^+}{1 - f_d}$$

be the MIR-function for $\leq$-base-inequalities with right-hand side d. The function φ_d is superadditive and nondecreasing with $\varphi_d(0) = 0$. Furthermore, if $d \notin \mathbb{Z}$ then

$$\lim_{t \searrow 0} \frac{\varphi_d(at)}{t} = \frac{a^-}{1 - f_d}$$

for all $a \in \mathbb{R}$ (see [15, §II.1.7]). Using the definitions and the relation $cf_{\frac{-a}{c}} = c - r(-a, c)$, it is easy to see that $F_{d,c}(a) = -r(d, c) \cdot \varphi_{-\frac{d}{c}}(-\frac{a}{c})$, which immediately gives the result. □

By Theorem 7.4 of [15, §II.1], functions with the properties described in Lemma 1 can be used to derive valid inequalities by applying them to the coefficients of valid base inequalities. The function $F_{d,c}$ can be seen as the $1/c$-MIR function for $\geq$-base-inequalities with right-hand side d and scaled by the factor $r(d, c)$. Notice that $F_{d,c}(a), \bar{F}_{d,c}(a)$ are integral if a, c, and d are integral. Moreover, $|F_{d,c}(a)| \leq |a|$ holds for all $a \in \mathbb{R}$, as shown in [16]. From a numerical point of view, both properties are desirable in a cutting plane algorithm. Similar subadditive and superadditive functions based on MIR have been considered in [1, 2, 3].

Proposition 1 ([1, 16]). *For any $t \in M$, let $F_t := F_{d_S^Q, c^t}$ and $\bar{F}_t := \bar{F}_{d_S^Q, c^t}$. The general flow-cutset inequality*

$$f^Q(\bar{A}_1^+) - f^Q(A_2^-) + \sum_{m \in M} F_t(c^m) x^m(A_1^+) + \sum_{m \in M} (c^m + F_t(-c^m)) x^m(A_2^-) \geq F_t(d_S^Q) \qquad (1)$$

is valid for P_{di}, whereas the following general flow-cutset inequality is valid for P_{bi} and P_{un}:

$$f^Q(\bar{A}_1^+) - f^Q(A_2^-) + \sum_{m \in M} F_t(c^m) x^m(E_1) + \sum_{m \in M} (c^m + F_t(-c^m)) x^m(E_2) \geq F_t(d_S^Q). \qquad (2)$$

Proof. By aggregating model inequalities and substituting $\bar{f}^Q(A_2^-) := \sum_{m \in M} c^m x^m(A_2^-) - f^Q(A_2^-)$, the following base inequality is valid for P_{di}:

$$f^Q(\bar{A}_1^+) + \bar{f}^Q(A_2^-) + \sum_{m \in M} c^m \big(x^m(A_1^+) - x^m(A_2^-)\big) \geq d_S^Q. \qquad (3)$$

Similarly, the following inequality is valid for P_{bi} and P_{un}:

$$f^Q(\bar{A}_1^+) + \bar{f}^Q(A_2^-) + \sum_{m \in M} c^m \big(x^m(E_1) - x^m(E_2)\big) \geq d_S^Q. \qquad (4)$$

Applying the function $F_t := F_{d_S^Q, c^t}$ to all coefficients of module variables and $\bar{F}_t := \bar{F}_{d_S^Q, c^t}$ to all coefficients of flow variables in (3) and (4) yields inequalities (1) and (2), respectively. Notice that $\bar{F}_t(1) = 1$. The proposition follows then by Lemma 1, [15, §II.1 Theorem 7.4] and resubstituting $\bar{f}^Q(A_2^-)$. □

Flow-cutset inequalities can easily be generalized to arc-dependent module sets. If $A_2^- = \emptyset$ (or $E_2 = \emptyset$) and $d_S^Q \geq 0$, the flow-cutset inequalities (1), (2) can be strengthened by rounding down all left-hand side coefficients to the value of the right-hand side. For ease of exposition, we leave out these generalizations and strengthenings in the following discussion. Necessary and sufficient conditions for (1) and (2) to define facets of their corresponding polyhedra can be found in [1] and [16], respectively.

Inequalities (1) and (2) generalize large classes of known valid inequalities for network design polyhedra. In particular, by choosing $A_2^- = \emptyset$ ($E_2 = \emptyset$) and $A_1^+ = A_S^+$ ($E_1 = E_S$) as well as $Q = K_S^+$, they reduce to the well-known cutset inequalities that contain only module variables [1, 4, 5, 13]. The general flow-cutset inequality (1) for DIRECTED models and $|M| = 1$ was introduced by Chopra et al. [7]. The first to study the multi-facility case with arbitrary capacity structure is Atamtürk [1] in the context of the directed cutset polyhedron. For the lifting functions ϕ_t^+, ϕ_t^- used to lift the inequalities from the single-facility to the multi-facility case in [1] it holds that $\phi_t^+(a) = F_t(a)$ and $\phi_t^-(a) = (c^m + F_t(-a))$ for $a \in \mathbb{R}$ [16]. Inequality (2) is the analogon of (1) for undirected networks. Magnanti et al. [11, 12, 13] study cutset inequalities for UNDIRECTED models with up to three modules for the special case of divisible capacities. Simple flow-cutset inequalities with $E_2 = \emptyset$ and $|M| = 2$ have been studied by Bienstock and Günlük [5].

In the light of [14], general flow-cutset inequalities are derived by *Aggregating*, *Substituting*, *Scaling* and *MIR*. In fact, the function F_t is responsible for the last two steps. It scales the base inequalities (3), (4) with $1/c^t$ and then applies MIR. There are also relations to the concept of simultaneous (subadditive) lifting [1, 2, 3, 10].

Separation

Given one of the network design polyhedra P_{di}, P_{bi} and P_{un} and a (fractional) point $\hat{p} = (\hat{f}, \hat{x})$, the separation problem for general flow-cutset inequalities reduces to the problem of simultaneously determining a nodeset $S \subset V$, a commodity subset $Q \subseteq K$, arc or edge subsets of the cut and a module $t \in M$ leading to a most violated inequality. Atamtürk [1] shows that this problem is already $\mathcal{NP}$-hard in the special case of a single point-to-point commodity. For general commodities, Barahona [4] proves that finding a most violated cutset inequality is $\mathcal{NP}$-hard by reduction from the *maxcut*-problem. For a fixed nodeset S, the complexity of simultaneously determining Q and A_1^+, A_2^- (E_1, E_2) is an open question. For fixed S and Q, however, suitable subsets of the cut arcs (cut edges) can be identified in linear time for every $t \in M$, as shown in [1].

Our generic separation heuristic is based on decomposing the separation problem. It combines ideas from [1, 5, 6, 8, 13]:

1. Contract the network and enumerate all cuts of the resulting partition. For every module $t \in M$ check the corresponding cutset inequality for violation.
2. Given a nodeset S, heuristically compute promising commodity subsets Q.
3. For given S and Q and for all $t \in M$, exactly determine subsets A_1^+, A_2^- or E_1, E_2 leading to the most violated general flow-cutset inequality of type (1) or (2).

The second and third step are only executed if no violated cutset inequalities have been found in the first step during several iterations. Among all identified violated inequalities, we add only those to the current LP which have a large distance to the point $\hat{p}$ and which are not too orthogonal to the objective function.

Finding a nodeset S To determine cutsets we generalize a heuristic proposed by Bienstock et al. [6] and Günlük [8]. Given weights w_a, w_e for the arcs and edges of the network, we iteratively shrink the arc (edge) with the largest weight (deleting loops). We apply this procedure until the shrunken graph has a predefined number of nodes (usually 3 or 4), and enumerate all cuts in that graph.

Given a fractional LP solution, let s_a, s_e be the slacks and π_a, π_e the dual values related to the capacity constraints of arc a or edge e. For the BIDIRECTED link model there are two capacity constraints for every edge $e = ij$ and we define slacks and duals to be $s_{ij}, s_{ji}, \pi_{ij}, \pi_{ji}$. We define the arc (edge) weights as

DIRECTED link model:	$w_a := s_a - \lvert\pi_a\rvert$	$a \in A$
BIDIRECTED link model:	$w_e := \min(s_{ij}, s_{ji}) - \max(\lvert\pi_{ij}\rvert, \lvert\pi_{ji}\rvert)$	$e = ij \in E$
UNDIRECTED link model:	$w_e := s_e - \lvert\pi_e\rvert$	$e \in E.$

The idea is to concentrate on cuts that have many arcs (edges) with small slack to raise the chance of obtaining a violated general flow-cutset inequality. Considering dual values provides a second sorting criterion for the common case that many capacity constraints are tight w. r. t. the current LP solution.

Finding a commodity subset Q In general no efficient way is known of computing a commodity subset that leads to a most violated general flow-cutset inequality. We concentrate on commodity subsets Q with $Q \subseteq K_S^+$ or by symmetry $Q \subseteq K_S^-$ (exchanging S and $V \backslash S$). Similar to [1, 5, 6], we iterate all singleton commodity subsets, some commodity subsets Q with $|Q| = 2$, and the whole commodity subset K_S^+ (K_S^-).

Finding subsets A_1^+ and A_2^- (or E_1 and E_2) Given $S \subset V$, $Q \subseteq K$ and $t \in M$, we derive a most violated general flow-cutset inequality for the DIRECTED link model [1] in linear time by defining

$$A_1^+ := \{a \in \delta^+(S) : \sum_{m \in M} F_t(c^m)\hat{x}_a^m < \hat{f}_a^Q\}, \qquad A_2^- := \{a \in \delta^+(S) : \sum_{m \in M} (c^m + F_t(-c^m))\hat{x}_a^m < \hat{f}_a^Q\}.$$

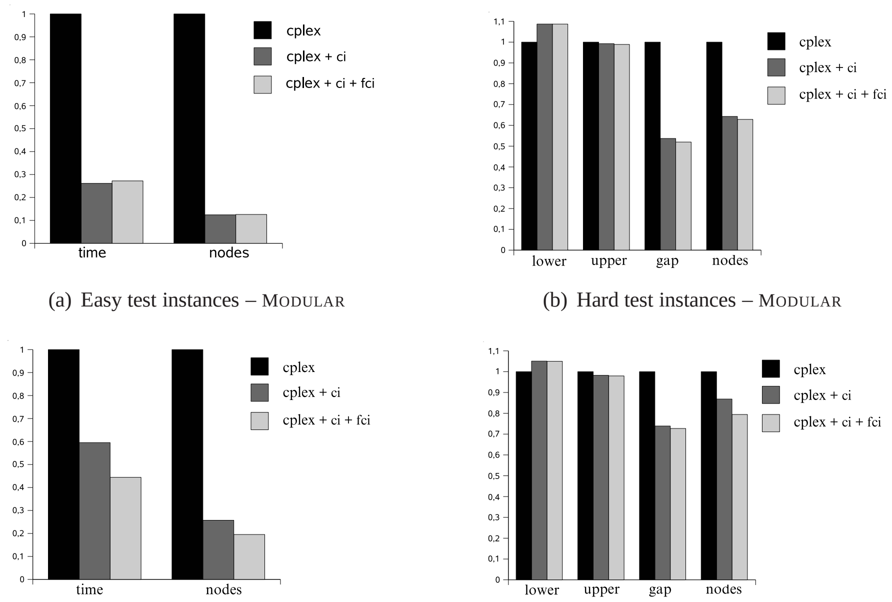
Similarly, for the BIDIRECTED and UNDIRECTED link models [16], we define

$$E_1 := \{e{=}ij \in \delta(S) : \sum_{m \in M} F_t(c^m)\hat{x}_e^m < \hat{f}_{ij}^Q\}, \qquad E_2 := \{e{=}ij \in \delta(S) : \sum_{m \in M} (c^m + F_t(-c^m))\hat{x}_e^m < \hat{f}_{ji}^Q\}.$$

Results

We tested the Branch & Cut approach on all instances of SNDlib [17] with the following model specifications: link models DIRECTED, BIDIRECTED or UNDIRECTED, link capacities MODULAR or EXPLICIT, no fixed-charge costs, continuous routing allowing all paths, no hop limits and no survivability. These are 54 problem instances in total (4 DIRECTED, 25 BIDIRECTED, 25 UNDIRECTED). The number of nodes ranges from 10 to 65, the number of links ranges from 18 to 172, whereas the number of demands varies from 22 to 1869. The maximum number of modules was 11 for MODULAR link capacity models (27 cases) and 55 for EXPLICIT models (27 cases).

We used CPLEX 10.0 [9] in the default mode (*cplex*) and compared its performance to our algorithm that augments CPLEX by the presented separation routine. In the root node and in every fourth depth of the search tree our separation heuristic was applied to separate either cutset inequalities only (*cplex + ci*) or general flow-cutset inequalities (*cplex + ci + fci*) as described in the previous section. All computations were performed on a 3Ghz x86 machine with 2GB of RAM and a time limit of one hour.



(a) Easy test instances – MODULAR

(b) Hard test instances – MODULAR

(c) Easy test instances – EXPLICIT

(d) Hard test instances – EXPLICIT

Figure 2: Average improvements by adding general flow-cutset inequalities

The described separation scheme performed much better than default CPLEX. Figure 2 shows the average improvements made when adding general flow-cutset inequalities to the initial formulation, comparing the solution time (*time*), the number of visited nodes (*nodes*), the lower and upper bounds (*lower, upper*) and the final gap (*gap*). All values are given as a ratio to the values obtained by default CPLEX and averaged over all problem instances. We refer to 27 of the given problem instances as *easy* since our algorithm solved them to optimality. Default CPLEX could not solve 10 of these instances within the time or memory limit. Hence the actual improvement is even greater than shown in Figure 2(a) and 2(c). For the easy instances we could reduce the computation time by more than 70% for MODULAR problems and by more than 50% for EXPLICIT models. For the remaining *hard* instances, we were able to reduce the overall gap by almost 50% for MODULAR and by almost 30% for EXPLICIT problems.

It can be seen that cutset inequalities are responsible for most of the progress. The performance is only slightly better when the larger class of general flow-cutset inequalities is considered. For almost all of the instances we had no difficulties in finding violated general flow-cutset inequalities. The problem is to find the best ones. We believe that a more elaborate selection of cuts and commodity subsets might increase the impact of these inequalities.

Acknowledgment

This research has been supported by the German Ministry of Education and Research (BMBF) within the EIBONE project under contract number 01BP567.

References

[1] A. Atamtürk. On capacitated network design cut-set polyhedra. *Mathematical Programming*, 92:425–437, 2002.

[2] A. Atamtürk. On the facets of the mixed-integer knapsack polyhedron. *Mathematical Programming*, 98:145–175, 2003.

[3] A. Atamtürk. Sequence independent lifting for mixed-integer programming. *Operations Research*, 52: 487–490, 2004.

[4] F. Barahona. Network design using cut inequalities. *SIAM Journal on Optimization*, 6:823–837, 1996.

[5] D. Bienstock and O. Günlük. Capacitated network design – polyhedral structure and computation. *INFORMS Journal on Computing*, 8:243–259, 1996.

[6] D. Bienstock, S. Chopra, O. Günlük, and C. Y. Tsai. Minimum cost capacity installation for multicommodity network flows. *Mathematical Programming*, 81:177–199, 1998.

[7] S. Chopra, I. Gilboa, and S. T. Sastry. Source sink flows with capacity installation in batches. *Discrete Applied Mathematics*, 86:165–192, 1998.

[8] O. Günlük. A branch and cut algorithm for capacitated network design problems. *Mathematical Programming*, 86:17–39, 1999.

[9] CPLEX 10.0. *CPLEX 10.0 Reference Manual*. ILOG CPLEX Division, 889 Alder Avenue, Suite 200, Incline Village, NV 89451, USA, 2006. `http://www.ilog.com/products/cplex/`.

[10] Q. Louveaux and L. A. Wolsey. Lifting, superadditivity, mixed integer rounding and single node flow sets revisited. *4OR*, 1(3):173–207, 2003.

[11] T. L. Magnanti and P. Mirchandani. Shortest paths, single origin-destination network design and associated polyhedra. *Networks*, 33:103–121, 1993.

[12] T. L. Magnanti, P. Mirchandani, and R. Vachani. The convex hull of two core capacitated network design problems. *Mathematical Programming*, 60:233–250, 1993.

[13] T. L. Magnanti, P. Mirchandani, and R. Vachani. Modelling and solving the two-facility capacitated network loading problem. *Operations Research*, 43:142–157, 1995.

[14] H. Marchand and L. A. Wolsey. Aggregation and mixed integer rounding to solve MIPs. *Operations Research*, 49(3):363–371, 2001.

[15] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.

[16] C. Raack. Employing Mixed-Integer Rounding in Telecommunication Network Design. Diploma thesis, TU Berlin, December 2005. `http://www.zib.de/groetschel/students/masterstudents.html`.

[17] SNDlib. SNDlib 1.0 – Survivable Network Design data library. `http://sndlib.zib.de`, 2005.