Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

Dimitris Alevras   Martin Grötschel
Roland Wessäly

Cost–Efficient Network Synthesis from Leased Lines

Preprint SC 97-22 (June 1997)

Cost–Efficient Network Synthesis from Leased Lines

Dimitris Alevras * Martin Grötschel † Roland Wessäly ‡

Abstract

Given a communication demand between each pair of nodes of a network we consider the problem of deciding what capacity to install on each edge of the network in order to minimize the building cost of the network and to satisfy the demand between each pair of nodes.

The feasible capacities that can be leased from a network provider are of a particular kind in our case. There are a few so-called basic capacities having the property that every basic capacity is an integral multiple of every smaller basic capacity. An edge can be equipped with a capacity only if it is an integer combination of the basic capacities. We treat, in addition, several restrictions on the routings of the demands (length restriction, diversification) and failures of single nodes or single edges. We formulate the problem as a mixed integer linear programming problem and develop a cutting plane algorithm as well as several heuristics to solve it. We report on computational results for real world data.

Keywords: Telecommunication Network Design, Survivable Networks, Network Capacity Planning, Cutting Plane Algorithm, Heuristics, Routing

Mathematics Subject Classification (1991): 90B12, 90C11, 90C27, 90C90, 94A99

1 Introduction

A telecommunication provider has to offer its service as cheap as possible and to keep its quality as high as possible. Installing and maintaining a network that achieves an appropriate balance between these conflicting goals is a difficult task. We describe here a problem of dimensioning a survivable telecommunication network that we encountered in a joint project with `E-Plus` Mobilfunk GmbH, one of the currently three mobile phone service providers in Germany: Given the nodes and the possible physical links of a telecommunication network, determine what capacity to install on the links to satisfy the demands and certain survivability requirements.

*IBM Corporation, Philadelphia, USA, alevras@us.ibm.com

†Konrad-Zuse-Zentrum für Informationstechnik, Berlin, Germany, groetschel@zib.de

‡Konrad-Zuse-Zentrum für Informationstechnik, Berlin, Germany, wessaely@zib.de

This problems has many versions; see, e.g., [1, 2, 3, 4, 7, 8, 10, 11, 14, 15, 16, 18] to mention a few relevant references. Of course, the link capacities must be chosen in a such way that all demands can be satisfied. An important design aspect is the protection of the network against component failures and the handling of such situations. Our partner `E-Plus` considers three options to set up a network with high survivability.

The first is called **reservation** and intends to equip the network with enough capacity such that, after the failure of a component, a certain percentage of each demand can still be satisfied. Utilizing this idea may require extensive rerouting of the communication paths in failure situations, and thus, additional management and maintenance efforts.

The second option is called **diversification** and has the goal to route every demand on several node-disjoint paths so that in case of a component failure, for each demand, not all paths are affected and the traffic can be directed through the surviving paths. By employing this concept, the network remains operational in failure situations without rerouting efforts.

Finally, the network designer may impose **length restrictions** on paths carrying communication traffic to avoid unacceptably long paths and to decrease the probability of a failure of a path component.

We have presented in [1] a mixed integer programming model, based on the work of Dahl and Stoer [7, 8], that provides a proper mathematical formulation of the concepts indicated above. We assumed in [1] that, for every link, the capacity can be chosen from a given finite set. This assumption on the capacities is very general. A number of practical situations can be solved adequately with this approach, see [1]. However, if many capacities are available the size of the resulting problem instance may become too large. To obtain reasonable upper and lower bounds on the optimal objective value (in acceptable running time) we had to reduce the number of available capacities considerably in a preprocessing step. Our computations result in satisfactory solutions, but we found later several cases where the optimal solution of our original problem was not feasible for the reduced model. Despite the limitations, this approach is justifiable and it is, in fact, still in use at `E-Plus`.

Nevertheless, we have been searching for ways to handle large numbers of capacities without reducing these artificially. Here we present such an approach that makes use of a special structure of the capacities available for lease that we encountered in many (but not all) practical problem instances. For example, *Deutsche Telekom* rents multiples of 2Mbit/s links (30 channels), multiples of 34Mbit/s links (480 channels) and multiples of 140Mbit/s links (1920 channels). In terms of channels these "basic capacities" have the property that each one is an integral multiple of each smaller one. The same is true, for instance, for the DS0, DS1, and DS3 facilities (offered in the US) which come with the "basic capacities" 1, 24, 672, respectively. The main topic of this paper is the treatment of survivable network design problems where each available link capacity is an integer combination of (a few) basic capacities with this special "divisibility property".

The algorithms for the solution of the models described in [1] and in this paper have been integrated in our network dimensioning tool DISCNET (DImensioning Survivable Cellular phone NETworks). The tool is in use at `E-Plus` for the annual transport network planning.

This rest of the paper is organized as follows: In the next section we formally define the problem and present the model. A high-level description of the algorithm is given in Section 3, while in Section 4, we describe the associated polytopes and classes of valid inequalities for these polytopes. In Section 5 we sketch our heuristics, and in Sections 6 and 7 we present computational results with real-world data and some conclusions, respectively.

2 The Model

Our network dimensioning problem is defined on two graphs on the same node set V, the **supply graph** $G = (V, E)$ and the **demand graph** $H = (V, D)$. The edge set E of the supply graph is the set of the **supply edges**, which represent the links that already exist or can be physically installed (in our case: leased from some supplier). The edge set D of the demand graph is the set of the **demand edges**. Such an edge is introduced whenever there is a communication demand between the end-nodes of this edge. With each demand edge $uv \in D$ we associate the following four parameters: the **communication demand** $d_{uv} \in \mathbb{N}$ between nodes u and v, the **reservation parameter** $\rho_{uv} \in [0, 1] \subseteq \mathbb{R}_+$, which is a lower bound on the fraction of the demand d_{uv} that must be served when a single node or single edge of the network fails, the **diversification parameter** $\delta_{uv} \in (0, 1] \subseteq \mathbb{R}_+$, which is an upper bound on the fraction of the demand d_{uv} that can be routed through a node or an edge in case all network components are operating (normal operating state), and the **path-length parameter** $\ell_{uv} \in \mathbb{N}$, which is an upper bound on the number of edges contained in a path that routes (part of) the demand d_{uv} in the normal operating state.

The **operating states** s we consider are the normal operating state $s = 0$ (all nodes and all supply edges are operating), the single node failures $s = v$ for each node $v \in V$, and the single supply edge failures $s = e$ for each edge $e \in E$. We denote by $G_s = (V_s, E_s)$ the supply graph and by $H_s = (V_s, D_s)$ the demand graph under operating state s. I.e., V_s is the set of nodes that are still functional in operating state s, and similarly, E_s is the set of available supply edges and D_s is the set of existing demands in operating state s.

We formulate the problem as a mixed integer linear programming problem with three types of variables.

A variable y_e is introduced for each supply edge $e \in E$. It denotes the capacity chosen for edge e. These variables are redundant, as we will see later, but they are introduced here for notational convenience.

The second type of variables are the "flow" variables $f(s, uv, P)$ which are introduced for each combination of an operating state s, a demand edge $uv \in D_s$, and a **valid path** P in G_s connecting the two end nodes u and v. If $s = 0$, a uv-path in G is valid if it contains at most ℓ_{uv} supply edges, while if $s \neq 0$ any uv-path in G_s is valid. The set of valid paths between u and v in operating state s is denoted by $\mathcal{P}(s, uv)$. In a solution of the problem, the value $f(s, uv, P)$ gives the amount of flow on the path P in G_s that satisfies part of the demand $\rho_{uv} d_{uv}$ in the operating states $s \neq 0$, or part of the total demand d_{uv} in the normal operating state $s = 0$.

With this notation we can formulate the continuous part of the mixed integer linear programming formulation of the problem.

$$\begin{aligned}
\sum_{uv \in D_s} \sum_{P \in \mathcal{P}(s,uv): e \in P} f(s, uv, P) &\leq y_e \qquad \forall s \text{ and } \forall e \in E_s && (1)\\
\sum_{P \in \mathcal{P}(0,uv)} f(0, uv, P) &= d_{uv} \qquad \forall uv \in D && (2)\\
\sum_{P \in \mathcal{P}(s,uv)} f(s, uv, P) &= \rho_{uv} d_{uv} \; \forall s \neq 0 \text{ and } \forall uv \in D_s && (3)\\
\sum_{P \in \mathcal{P}(0,uv): w \in P} f(0, uv, P) &\leq \delta_{uv} d_{uv} \; \forall uv \in D \text{ and } \forall w \in V \backslash \{u, v\} && (4)\\
f(0, uv, P) &\leq \delta_{uv} d_{uv} \; \forall uv \in D \text{ and } \forall P = \{uv\} && (5)\\
f(s, uv, P) &\geq 0 \qquad \forall s, \forall uv \in D_s \text{ and } \forall P \in \mathcal{P}(s, uv) && (6)
\end{aligned}$$

Constraints (1) bound, for each operating state s, the flow on each supply edge $e \in E$ by the capacity y_e of the edge. The equations (2) and (3) impose the demand requirements in the normal operating state $s = 0$ and in the failure states $s \neq 0$. Constraints (4) and (5) bound from above, for every demand edge $uv \in D$, the flow through all edges and nodes (other than the nodes u and v) by $\delta_{uv} d_{uv}$. This implies a diversification of the corresponding flow on several disjoint paths. Finally, (6) are the nonnegativity constraints.

Before we introduce the variables to decide the capacities of the supply edges, let us define a property of a set of numbers that is used to characterize the capacity structure. This property will frequently be used when we deal with valid inequalities for the associated polyhedra.

Property 2.1 (Divisibility)
Let $M = \{m_1, \ldots, m_k\} \subseteq \mathbb{N}$ with $m_1 \leq m_2 \leq \cdots \leq m_k$. We say that M has the divisibility property if the coefficients $\frac{m_{i+1}}{m_i}$ are integral for all $i = 1, \ldots, k-1$.

The capacities to be installed have the following particular structure. We are given a set $T = \{\tau_1, \ldots, \tau_n\}$ of **technologies**, one for each different type of line that can be installed on a link. With each technology $\tau \in T$ we associate a positive **basic capacity** M^τ and the edge dependent installation **costs** (which include a fixed cost and a variable cost which varies with the length of a link). We assume that the basic capacities satisfy Property 2.1 and we refer to the smallest basic capacity M^{τ_1} as the **unit capacity**.

Now, the third type of variables can be defined. For each supply edge $e \in E$, we are given a set $t(e) \subseteq T$ of available technologies. The capacities that can be installed on edge e are integer combinations of the basic capacities $M^\tau, \tau \in t(e)$, of the available technologies plus an additional **free capacity** M_e^0. We model this structure by introducing, for every supply edge $e \in E$ and every available technology $\tau \in t(e)$, a nonnegative integer variable x_e^τ to denote the integral multiple of M^τ. The variables x_e^τ may be restricted by an upper bound u_e^τ. For each $\tau \in t(e)$ we denote by K_e^τ the cost of installing one unit of the basic capacity M^τ on supply edge $e \in E$.

The objective is to minimize the total cost of installing the necessary capacities on the edges of the supply graph. This is formulated as

$$\min \sum_{e \in E} \sum_{\tau \in t(e)} K_e^\tau x_e^\tau \ .$$

The constraints that must be satisfied in addition to $(1), \ldots, (6)$ are the nonnegativity and, if required, the upper bound constraints

$$0 \leq x_e^\tau \leq u_e^\tau \qquad \text{and} \qquad x_e^\tau \in \mathbb{N} \qquad \forall e \in E \text{ and } \forall \tau \in t(e) \ , \tag{7}$$

where the capacity y_e of a supply edge $e \in E$ is

$$y_e = M_e^0 + \sum_{\tau \in t(e)} M^\tau x_e^\tau . \tag{8}$$

As we mentioned before, the variables y_e – and thus the constraints (8) – are not used explicitly in the linear program. They are rather calculated from these equations given a vector x.

A **feasible solution** is a vector (x, y, f) that satisfies the constraints (1) - (8). For our purposes, we assume that there exists a feasible solution. This assumption implies the existence of at least $\lceil \frac{1}{\delta_{uv}} \rceil$ node-disjoint length-restricted paths for each demand edge $uv \in D$. Note, that the problem to decide whether there exist at least k node-disjoint paths of length at most ℓ is $\mathcal{NP} - complete$ (see, e.g., Garey and Johnson [9]). For the sizes of the problem instances we consider, the framework of Bley [5] suffices to find such paths, if they exist, in reasonable running times.

3 Algorithmic Approach

We solve the problem presented in the previous section with a cutting plane algorithm followed by linear programming based heuristics. Figure 1 shows the flow chart of the algorithm. The algorithm consists of three main parts:

(i) the feasibility problem (FP) (that decides whether a given capacity vector admits feasible routings in all operating states),

(ii) the cutting plane part (that calculates a lower bound on the optimal value of a feasible solution), and,

(iii) the heuristic algorithms (that produce "good" feasible solutions).

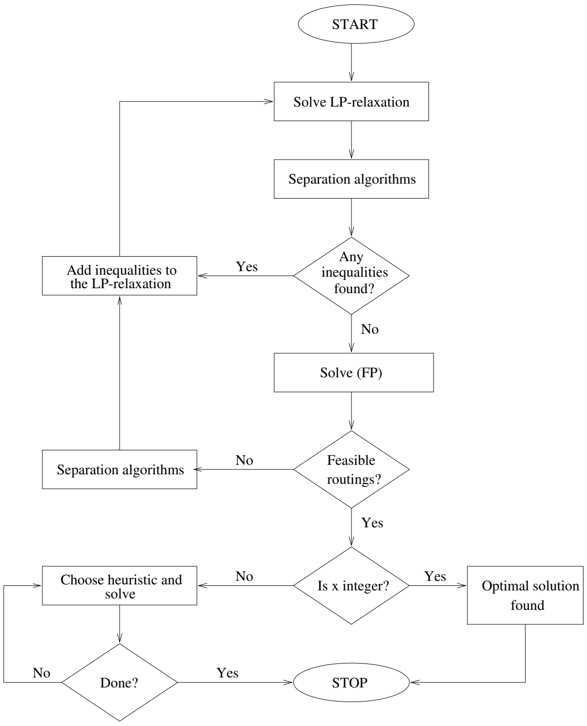

Figure 1: Flow chart of the algorithm.

The feasibility problem (FP) is defined by the constraints (1), ... , (6). This problem decomposes into one multicommodity-flow problem for each operating state. To solve each individual multicommodity-flow problem we apply a column generation approach suggested by Minoux and also used by Dahl and Stoer (see [16], [7] and [1]).

In every iteration of the cutting plane part we solve an LP-relaxation of the integer program in the decision variables x, which contains the constraints (7) and a subset of the known valid inequalities for the polytope of feasible x-vectors, see Section 4. The valid inequalities (cutting planes) we use are

(i) partition inequalities introduced by Pochet and Wolsey [17],

(ii) strengthened partition inequalities, see Section 4,

(iii) strengthened metric inequalities, see Section 4, and

(iv) diversification-cut inequalities, see Section 4.

One iteration of the cutting plane algorithm is as follows: Given the solution of the current LP-relaxation we use separation algorithms (sketched in Section 4) to find valid inequalities that are violated by this solution. We add all violated inequalities found to the LP-relaxation and resolve it. If we cannot generate any violated inequality we calculate via (8) a (possibly fractional) capacity vector. For this capacity vector we decide (FP), that is, we test whether there exist feasible routings in all operating states. If not, we can identify a violated metric inequality. From this we derive violated inequalities in x-variables, add these and start the next iteration with the augmented LP-relaxation.

Eventually, if the (possibly fractional) capacity vector is feasible, there are two possibilities. If the x-variables are integer, we have found an optimal solution and we are done. Otherwise, we resort to various heuristic algorithms (see Section 5) to obtain "good" integer solutions. The heuristic algorithms are from two classes of heuristics. The first class consists of primal improvement heuristics, whereas the second class consists of a kind of dual heuristics that follow one branch of a branch-and-cut tree.

The cutting plane phase provides a lower bound and the best heuristic solution provides an upper bound on the optimal solution value (the minimum cost). Thus, we have a guaranteed quality of the solutions.

4 Related Polyhedra and Valid Inequalities

In this section we describe the polyhedra associated with the model presented in Section 2 and valid inequalities for these polyhedra. We investigate two polyhedra: (i) the convex hull of feasible solutions in terms of x-variables, and, (ii) a polyhedron in the y-variables that can be viewed as a relaxation of the polytope in the x-variables.

4.1 Two Polyhedra

The convex hull of all feasible solutions in x-variables is

$$X \quad := \quad \mathrm{conv}\{x \mid \exists (y,f) \text{ such that } (x,y,f) \text{ satisfies } (1),\dots,(8)\}, \tag{9}$$

and the relaxation to the continuous y-variables

$$Y \quad := \quad \{y \mid \exists f \text{ such that } (y,f) \text{ satisfies } (1),\dots,(6)\}. \tag{10}$$

There is an obvious relation between these two polyhedra. If $\sum_{e\in E} a_e y_e \geq \alpha$ is valid for Y, then $\sum_{e\in E} a_e \sum_{\tau\in t(e)} M^\tau x_e^\tau \geq \alpha - \sum_{e\in E} a_e M_e^0$ is valid for X. To keep the exposition simple, we assume w.l.o.g. throughout the remainder of this section that all free capacities M_e^0 are zero.

4.2 Valid inequalities for the polyhedron Y

The polyhedron Y is the set of solutions to our feasibility problem (FP), defined by the inequalities (1), ... , (6). Obviously, this problem decomposes into one continuous multicommodity-flow problem for each operating state. Iri [12], and Kakusho and Onaga [13] independently characterized the solutions of continuous multicommodity-flow problems using the so-called *metric inequalities*. In our case, a generalization of these metric inequalities describes the polyhedron Y (see also Dahl and Stoer [7]).

Metric inequalities

Let $y = (y_e)_{e\in E}$ be a vector of capacities. For every operating state $s \neq 0$, the corresponding multicommodity-flow problem $(MCFP_s)$ is defined by the inequalities (1), (3) and (6) (for this operating state s). In the normal operating state the multicommodity-flow problem $(MCFP_0)$ is defined by the inequalities (1), (2), (4), (5) and (6), for $s = 0$. Necessary and sufficient conditions for the feasibility of y are given in the following two theorems.

Theorem 4.1 ([12], [13])
A capacity vector y is feasible for $(MCFP_s)$, $s \neq 0$, if and only if

$$\sum_{e\in E_s} \mu_e y_e \geq \sum_{uv\in D_s} \pi_{uv} d_{uv} \tag{11}$$

for all $\mu_e \geq 0$ $(e \in E)$, where, for every $uv \in D_s$, π_{uv} is the value of a shortest uv-path in G_s with respect to the edge weights μ_e.

In our case the multicommodity-flow problems are more complicated in the normal operating state, because of the diversification and the path-length constraints. In this case, the above result can be modified as follows.

Theorem 4.2 ([7])
A capacity vector y is feasible for $(MCFP_0)$ if and only if

$$\sum_{e\in E} \mu_e y_e \geq \sum_{uv\in D} d_{uv}\pi_{uv} - \sum_{uv\in D} (\delta_{uv} d_{uv} \gamma_{uv}^{uv} + \sum_{w\in V\setminus\{u,v\}} \delta_{uv} d_{uv} \gamma_{uv}^{w}) \tag{12}$$

for all $\mu_e \geq 0$ $(e \in E)$, $\gamma_{uv}^w \geq 0$ $(uv \in D, w \notin \{u,v\})$, and $\gamma_{uv}^{uv} \geq 0$ $(uv \in D, uv \in E)$. π_{uv} is defined as follows: Given $uv \in D$, we assign to each edge $e \in E\setminus\{uv\}$ the weight μ_e, to edge uv (if it is contained in E) the weight $\mu_e + \gamma_{uv}^{uv}$, and to each node $w \in V\setminus\{u,v\}$ the weight γ_{uv}^w. Then π_{uv} is the value of a shortest among all uv-paths in G with at most ℓ_{uv} edges.

Inequalities (11) and (12) are called **metric inequalities**. We often write a metric inequality as $\sum_{e\in F} \mu_e y_e \geq d$, where $F := \{e \in E \mid \mu_e > 0\}$ and d is the right hand side in (11) or (12).

Since the feasibility problem (FP) decomposes into one multicommodity-flow problem for each operating state, the Theorems 4.1 and 4.2 together yield necessary and sufficient conditions to decide (FP) for a given capacity vector y.

Cut inequalities

A special case of a metric inequality is a cut inequality. Given $W \subseteq V$, define $\mu_e = 1$ for every $e \in \delta_G(W) := \{wz \in E : w \in W,\ z \in V \backslash W\}$, and $\mu_e = 0$ otherwise. Furthermore, we set $\gamma_{uv}^w = \gamma_{uv}^{uv} = 0$ for every $uv \in D$ and every $w \in V \backslash \{u, v\}$. Then $\pi_{uv} = 1$ for every $uv \in \delta_H(W)$, and $\pi_{uv} = 0$ otherwise, are – under appropriate connectivity assumptions – the shortest path lengths from u to v with respect to edge weights μ. Then inequality (12) reads as follows:

$$\sum_{e \in \delta_G(W)} y_e \geq \sum_{uv \in \delta_H(W)} d_{uv} \ . \tag{13}$$

These inequalities are called **cut inequalities**.

4.3 Valid inequalities for the polyhedron X

Based on valid inequalities for Y we now derive two classes of valid inequalities for X. The first class, the *strengthened metric inequalities*, is just the result of a divide-and-round procedure. The second class, the *partition inequalities*, describe the relaxation given by a cut inequality or, more generally, by the relaxation given by a metric inequality with coefficients that satisfy Property 2.1. If we are to consider failure situations we derive stronger inequalities, the *strengthened partition inequalities*. A third class of inequalities valid for X, not based on a valid inequality for Y, is the class of *diversification-cut inequalities*. If survivability is implemented by setting some diversification parameters to a value smaller than one then this class has proven to be very useful in the lower bound calculation (see Section 6).

Strengthened metric inequalities

Let $\sum_{e \in F} \mu_e y_e \geq d$ be a metric inequality. Using equality (8) we substitute y-variables by x-variables and get the inequality

$$\sum_{e \in F} \mu_e \sum_{\tau \in t(e)} M^\tau x_e^\tau \geq d \ , \tag{14}$$

which is apparently valid for X. To this inequality we apply a divide-and-round procedure to get a stronger inequality.

Proposition 4.3 *Let $\sum_{e \in F} \mu_e y_e \geq d$ be a metric inequality and g be the greatest common divisor of the coefficients, i.e., $g := gcd\{\mu_e M^\tau \mid e \in F, \tau \in t(e)\}$. Then the inequality*

$$\sum_{e \in F} \sum_{\tau \in t(e)} \min\{\frac{\mu_e M^\tau}{g}, \lceil \frac{d}{g} \rceil\} \ x_e^\tau \geq \lceil \frac{d}{g} \rceil \tag{15}$$

is valid for X.

Proof. Divide the coefficients of (14) by g and round up the right-hand-side of the inequality. The resulting inequality is due to the integrality of every feasible solution valid for X. Finally, take for each coefficient the minimum of this coefficient and the right-hand-side. ■

In the course of our cutting plane procedure we try to overcome the strengthening of metric inequalities because these inequalities are very dense (one positive coefficient for each $e \in F$ and each available technology of these edges) and, more importantly, their "wild" coefficients cause numerical instabilities.

Given a metric inequality for the normal operating state we can further strengthen it, in particular, if the reservation parameter is close to 1.0 for all demands.

Proposition 4.4 *Let $\sum_{e\in F} \mu_e y_e \geq \sum_{uv\in D} \pi_{uv} d_{uv}$ be a metric inequality and g be the greatest common divisor of the coefficients, i.e., $g := gcd\{\mu_e M^\tau \mid e \in F, \tau \in t(e)\}$. Then the inequality*

$$\sum_{e\in F} \sum_{\tau\in t(e)} \min\{\frac{\mu_e M^\tau}{g}, \lceil\frac{d}{g}\rceil\}\ x_e^\tau \geq \lceil \frac{|F|}{|F|-1} \cdot \lceil\frac{d}{g}\rceil\rceil \tag{16}$$

is valid for X, where $d := \sum_{uv\in D} \pi_{uv}\rho_{uv} d_{uv}$.

Proof. Choose $h \in F$ and let π_{uv}^h be, for every $uv \in D$, the value of a shortest uv-path in G_h w.r.t. μ. Then

$$\sum_{e\in F\setminus\{h\}} \mu_e y_e \geq \sum_{uv\in D} \pi_{uv}^h \rho_{uv} d_{uv} \geq \sum_{uv\in D} \pi_{uv}\rho_{uv} d_{uv} \qquad (= d)\ , \tag{17}$$

since $\pi_{uv}^h \geq \pi_{uv}$ for every $uv \in D$. If we sum inequality (17) over all $h \in F$, we get

$$(|F|-1|) \sum_{e\in F} \mu_e y_e \geq |F| \sum_{uv\in D} \pi_{uv}\rho_{uv} d_{uv}\ . \tag{18}$$

Finally, the result follows with the same arguments as in the proof of Proposition 4.3. ■

We call inequalities of type (15) or (16) **strengthened metric inequality**.

Partition inequalities

Again, let $\sum_{e\in F} \mu_e y_e \geq d$ be a metric inequality. Furthermore, let us assume that the coefficients $M := \{\mu_e M^\tau \mid e \in F, \tau \in t(e)\}$ satisfy Property 2.1. This assumption is satisfied, for instance, if the metric inequality is a cut inequality. Ordering and renaming we rewrite the coefficients M as $M = \{m_1, \ldots, m_n\}$ with $0 \leq m_1 \leq m_2 \leq \cdots \leq m_n$. Renaming the x-variables accordingly, the inequality $\sum_{e\in F}\mu_e \sum_{\tau\in t(e)} M^\tau x_e^\tau \geq d$ reads as $\sum_{i=1}^n m_i x_i \geq d$ and gives rise to the knapsack cover polytope

$$Q := \text{conv}\{x \in \mathbb{Z}_+^n \mid \sum_{i=1}^n m_i x_i \geq d\} . \tag{19}$$

Q is apparently a relaxation of X, i.e., $X \subseteq Q$. Introducing so-called partition inequalities, Pochet and Wolsey provide a complete linear characterization of Q in [17]. For the sake of completeness we briefly describe these inequalities here.

To introduce partition inequalities we define $r := \max\{i \mid m_i < d, 1 \leq i \leq n\}$ and we partition M into t consecutive blocks $\{i_1, \ldots, j_1\}, \ldots, \{i_t, \ldots, j_t\}$ such that $i_1 = 1, i_t \leq r, j_t = n$ and $i_k - 1 = j_{k-1}$ for $k = 2, \ldots, t$. Furthermore, we set $d_t := d$ and define recursively the coefficients $\kappa_k := \lceil \frac{d_k}{m_{i_k}} \rceil$ and the remaining demands $d_{k-1} := d_k - (\kappa_k - 1)m_{i_k}$, for $k = t, \ldots, 1$.

Proposition 4.5 [17] *The inequality*

$$\sum_{k=1}^t (\prod_{s=1}^{k-1} \kappa_s) \sum_{i=i_k}^{j_k} \min\{\frac{m_i}{m_{i_k}}, \kappa_k\} x_i \geq \prod_{s=1}^t \kappa_s \tag{20}$$

is valid for Q.

Inequalities (20) are called **partition inequalities** and suffice to describe Q.

Theorem 4.6 [17] *Q is the solution set of the system* (20) *of partition inequalities and the nonnegativity constraints.*

Strengthened partition inequalities

If we are to consider failure situations we can strengthen partition inequalities. Let $W \subseteq V$. Instead of the cut inequality $\sum_{e\in\delta_G(W)} y_e \geq \sum_{uv\in\delta_H(W)} d_{uv}$ we now consider the weaker version of the cut inequality

$$\sum_{e\in\delta_G(W)} y_e \geq \sum_{uv\in\delta_H(W)} \rho_{uv} d_{uv} . \tag{21}$$

Let

$$P := \{x \in \mathbb{N}^{|T(\delta_G(W))|} \mid \sum_{e\in\delta_G(W)} \sum_{\tau\in t(e)} M^\tau x_e^\tau \geq \sum_{uv\in\delta_H(W)} \rho_{uv} d_{uv}\}$$

where $T(\delta_G(W)) := \bigcup_{e\in\delta_G(W)} t(e)$. Now, assume that the partition inequality $\sum_{k=1}^t (\prod_{s=1}^{k-1}\kappa_s) \sum_{i=i_k}^{j_k} \min\{\frac{m_i}{m_{i_k}}, \kappa_k\} x_i \geq \prod_{s=1}^t \kappa_s$ is valid for P. Then we can derive the stronger inequality

$$\sum_{k=1}^t (\prod_{s=1}^{k-1} \kappa_s) \sum_{i=i_k}^{j_k} \min\{\frac{m_i}{m_{i_k}}, \kappa_k\} x_i \geq \lceil \frac{|\delta_G(W)|}{|\delta_G(W)| - 1} \prod_{s=1}^t \kappa_s \rceil . \tag{22}$$

Inequalities of this type are called **strengthened partition inequalities**. The inequalities (22) are stronger than partition inequalities, if the reservation parameter equals 1.0 for all demands in the cut.

Proposition 4.7 *Inequalities* (22) *are valid for* X.

Proof. The result follows as in the proof of Proposition 4.4, if we sum up the respective partition inequalities for all edge failures $f \in \delta_G(W)$.

■

Diversification-cut inequalities

The third class of inequalities is the class of diversification-cut inequalities.

Proposition 4.8 *Let* $W \subseteq V$ *and* M^{τ_1} *be the unit capacity. Then the inequality*

$$\sum_{e \in \delta_G(W)} \sum_{\tau \in t(e)} \min\{\frac{M^\tau}{M^{\tau_1}}, \lceil \frac{\alpha}{M^{\tau_1}} \rceil\} x_e^\tau \quad \geq \quad \lceil \frac{d}{M^{\tau_1}} \rceil \tag{23}$$

is valid for X, *where* $d := \sum_{uv \in \delta_H(W)} d_{uv}$ *and* $\alpha := \sum_{uv \in \delta_H(W)} \delta_{uv} d_{uv}$.

Proof. Let x be an integral feasible point for X, choose $e \in \delta_G(W)$ and let f be a feasible flow vector w.r.t. the capacities $y_e = \sum_{\tau \in t(e)} M^\tau x_e^\tau$. The diversification parameters imply (see (4), (5)) that

$$\sum_{uv \in \delta_H(W)} \sum_{P \in \mathcal{P}(0,uv): e \in P} f(0, uv, P) \leq \alpha \ ,$$

and thus

$$\sum_{uv \in \delta_H(W)} \sum_{P \in \mathcal{P}(0,uv): e \in P} f(0, uv, P) \leq \min\{\alpha, y_e\} \leq \sum_{\tau \in t(e)} \min\{M^\tau, \alpha\} x_e^\tau \ .$$

Summing up over all $e \in \delta_G(W)$ we get

$$\begin{aligned} \sum_{e \in \delta_G(W)} \sum_{\tau \in t(e)} \min\{M^\tau, \alpha\} x_e^\tau &\geq \sum_{e \in \delta_G(W)} \sum_{uv \in \delta_H(W)} \sum_{P \in \mathcal{P}(0,uv): e \in P} f(0, uv, P) \\ &= \sum_{uv \in \delta_H(W)} \sum_{e \in \delta_G(W)} \sum_{P \in \mathcal{P}(0,uv): e \in P} f(0, uv, P) \\ &\geq \sum_{uv \in \delta_H(W)} d_{uv} \ = \ d \ . \end{aligned}$$

Now, divide by the unit capacity. Finally, the integrality of x proves the proposition. ■

Inequalities (23) are called **diversification-cut inequalities**. Clearly, one can improve (23) if for all supply edges $e \in \delta_G(W)$ the available technologies $t(e)$ do not contain the technology τ_1. Then, the unit capacity M^{τ_1} can be substituted by the smallest basic capacity over all supply edges in the cut.

4.4 Identification of violated inequalities

We now sketch, for the classes of valid inequalities defined above, the separation algorithms we use to identify inequalities violated by the solution of an LP-relaxation during the cutting plane algorithm. For the classes of metric and partition inequalities we know exact separation algorithms. We separate diversification-cut inequalities heuristically.

Separation of (strengthened) metric inequalities

We can solve the separation problem for the class of metric inequalities exactly, i.e., we find a violated metric inequality, provided it exists. Moreover, we find a violated metric inequality in polynomial time, using linear programming, whenever we test the feasibility of a capacity vector which turns out to be infeasible. For details of this approach see Minoux [16], Stoer and Dahl [7], and [1].

We store identified metric inequalities and identified cut inequalities in two different cutting plane pools. Whenever we cannot find any other violated inequality (partition inequality, diversification-cut inequality) we check whether there is a metric inequality with associated violated strengthened metric inequality in the pool.

Separation of (strengthened) partition inequalities

For any cut inequality, or any inequality valid for Y satisfying the divisibility property (Property 2.1) we are given a relaxation Q of X defined by this inequality as in (19). For those relaxations we solve the separation problem for partition inequalities exactly with an algorithm proposed by Pochet and Wolsey. We refer the reader to [17] for details.

The identification of violated strengthened partition inequalities is performed with almost the same separation algorithm. We only have to initialize it differently and to calculate a different right-hand-side.

Separation of diversification-cut inequalities

We generate a diversification-cut inequality, whenever we find a violated cut inequality. This is the case either when we test the connectivity of the graph defined by the supply edges with positive capacity, or when we identify a cut inequality in the feasibility test of a capacity vector.

5 Heuristic Algorithms

We use two classes of heuristics to compute integer feasible solutions: one class consists of primal improvement heuristics and the other class of dual heuristics. To reduce the running time we apply a preprocessing procedure before we employ any of the heuristics. First we describe the preprocessing and then the two classes of heuristics in more detail.

5.1 Preprocessing

The preprocessing is fixing of variables. Based on various criteria we fix a minimum capacity on promising supply edges, or we exclude supply edges which are not necessary to build a feasible network, and we exclude capacities that we consider too expensive. To decide the supply edges for which we wish to fix a minimum capacity we use a mixture of the following criteria:

- Fix the edges in the cheapest path connecting the end-nodes of the demand edges with the biggest demand values,
- fix the supply edges with the biggest fractional capacity, after the lower bound calculation, and,
- fix edges to realize a sufficiently connected network.

Depending on the particular problem instance we decide how many supply edges to fix, and to which minimum capacity.

To exclude supply edges or expensive capacities we use a mixture of the following criteria:

- We exclude the longest supply edges with capacity zero, if the lower bound calculation was performed. Note, since the fractional solution after the lower bound calculation is feasible, we know that the remaining supply edges permit an integer feasible solution.
- We order the capacities w.r.t. their costs and remove a certain number of the most expensive ones, if we can still guarantee an integer feasible solution.

We implemented both the fixing of minimum capacities, and the removal of supply edges and capacities through constraints that we add to the LP-relaxation.

5.2 Decrease Heuristics

The decrease heuristics are primal heuristics, i.e., we start with a feasible capacity vector and try to reduce the capacity of its components (supply edges) keeping it feasible. A feasible capacity vector is obtained either by rounding up the capacity y_e of each edge $e \in E$ – as calculated from the solution to the current LP-relaxation via (8) – to the next bigger feasible capacity, or by the solution of an increase heuristic.

To keep the number of possibilities small we first select a technology, then the supply edges that can be reduced potentially, and then, we choose among these edges according to three different criteria the particular supply edge for which the capacity reduction will be tried.

The technologies are selected either in increasing or decreasing order of their capacities. Given a technology τ the three criteria to select the supply edge are:

- $\max\{K_e^\tau \mid e \in E\}$,

- $\max\{\frac{K_e^\tau}{y_e} \mid e \in E\}$,
- $\max\{\frac{K_e^\tau}{y_e} * 0.9^{k_e} \mid e \in E\}$, where k_e is the number reductions we already applied to edge e.

5.3 Increase Heuristics

The increase heuristics are dual heuristics. While the (primal) decrease heuristics maintain a feasible integer solution in every iteration, the increase heuristics maintain a feasible fractional solution in every iteration and terminate as soon as an integer feasible solution is constructed.

In every iteration we choose one supply edge among those with at least one fractional x-variable and fix some of the x-variables of the chosen supply edge to an integer value. Then we employ the cutting plane procedure, as described in Section 3, to calculate a lower bound on the optimal solution value of the restricted problem. Since we can guarantee a feasible (fractional) solution at the end of the cutting plane procedure, we have found an integer feasible solution as soon as all x-variables are integer at the end of one iteration. We apply three different criteria to choose the supply edge and two criteria to decide the fixing of the x-variables. This way we get six different increase heuristics.

Let $\overline{x} = (\overline{x}_e^\tau), e \in E, \tau \in t(e)$, be given and define $frac(\overline{x})$ as the set of supply edges with at least one fractional variable, i.e., $frac(\overline{x}) = \{e \in E \mid \exists \tau \in t(e) \text{ with } \overline{x}_e^\tau \neq \lfloor \overline{x}_e^\tau \rfloor\}$. Furthermore, let $\lceil y_e \rceil$ be defined as the smallest feasible capacity bigger than $y_e = M_e^0 + \sum_{\tau \in t(e)} M^\tau \overline{x}_e^\tau$. Then there exists a cheapest integer vector z (satisfying (7)) which yields the capacity $\lceil y_e \rceil$, i.e., $\lceil y_e \rceil = M_e^0 + \sum_{\tau \in t(e)} M^\tau z_e^\tau$. The criteria to choose the supply edges are:

1. $\max\{\overline{x}_e^\tau \mid e \in frac(\overline{x})\}$,
2. $\min\{\sum_{\tau \in t(e)} K_e^\tau z_e^\tau / \sum_{\tau \in t(e)} K_e^\tau \overline{x}_e^\tau \mid e \in frac(\overline{x})\}$,
3. $\min\{\sum_{\tau \in t(e)} K_e^\tau z_e^\tau - \sum_{\tau \in t(e)} K_e^\tau \overline{x}_e^\tau \mid e \in frac(\overline{x})\}$.

Given a supply edge $\overline{e} \in frac(\overline{x})$ we use either a greedy or, what we call, a conservative strategy to fix the x-variables. In more detail:

1. greedy: set $\overline{x}_{\overline{e}}^\tau = z_{\overline{e}}^\tau$ for all $z_{\overline{e}}^\tau > 0$,
2. conservative: set $\overline{x}_{\overline{e}}^\tau = z_{\overline{e}}^\tau$ for the maximum $M^\tau, \tau \in t(\overline{e})$ with $z_{\overline{e}}^\tau > 0$.

Both the greedy and the conservative fixing of x-variables can be implemented in terms of linear inequalities. These are added to the LP-relaxation. We remove these additional inequalities after an increase heuristic is finished.

Post-processing for increase heuristics

The result of an increase heuristic is not necessarily a local optimum, however, it is often either a local optimum or a "good" starting point for a decrease heuristic. In a typical run we try all decrease heuristics after an increase heuristic.

Both classes of heuristics are not very sophisticated. Currently we work on combinatorial heuristics that are independent from the final fractional solution of a cutting plane procedure. Nevertheless, we have empirical evidence that the implemented heuristics achieve "good" solutions (we used a branch-and-bound implementation to calculate optimal solutions for small instances).

6 Computational Results

In this section we present computational results for different problem instances supplied by `E-Plus`, with various settings for the demand related parameters, i.e., diversification, reservation and path length restriction.

We used four networks to test the program, the characteristics of which are given in Table 1. The number of nodes varies from 11 to 17, the number of supply edges from 34 to 53, and the number of demand edges from 24 to 106. The unit of the demand value is a channel (64 kbit/s). The available capacities for each supply edge are multiples of 30 channels (2 Mbit/s), multiples of 480 channels (34 Mbit/s) and multiples of 1920 channels (140 Mbit/s), and any nonnegative integer combination of these three basic capacities.

Name	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	range of demands
Network 1	11	34	24	95 – 384
Network 2	12	53	28	34 – 480
Network 3	14	39	82	30 – 360
Network 4	17	62	106	30 – 570

Table 1: Characteristics of the test problems.

We chose the following parameter settings. The length restriction takes the two values 3 and ∞ (no length restriction). For each length we set four different diversification/reservation pairs. The different parameter settings together with the names of the problems are given in Table 2.

Name	d1r0	d1r50	d1r100	d50r0
Diversification	1.0	1.0	1.0	0.5
Reservation	0.0	0.5	1.0	0.0

Table 2: Diversification and reservation parameter settings.

We run all tests on a SUN Ultra-1 with 512 MB main memory. As solver for linear programs we used the callable library of CPLEX 3.0 ([6]).

The total time reported in the tables corresponds to a *complete run* of the program that consists of the calculation of the lower bound (cutting-plane part), the execution of six decrease heuristics, and the execution of six increase heuristics. Each of the increase heuristics is followed by a run of all decrease heuristics.

		Lower bound	Time (min::sec)	Gap (%)
Network 1	d1r0	7415650.30	0:02.00	37.78
	d1r50	7408175.04	4:33.19	45.46
	d1r100	10088959.59	7:03.33	28.91
	d50r0	9397430.06	0:48.98	45.25
Network 2	d1r0	9059085.28	0:18.51	46.46
	d1r50	9099417.59	22:10.53	41.53
	d1r100	11682626.60	41:28.99	40.05
	d50r0	11256624.05	2:53.20	44.85
Network 3	d1r0	11920116.38	1:02.02	65.28
	d1r50	12252879.88	129:54.60	59.10
	d1r100	15023932.47	140:42.67	65.65
	d50r0	14581646.43	22:36.53	50.35
Network 4	d1r0	16061849.77	4:04.32	72.98
	d1r50	16191917.98	720:23.41	73.52
	d1r100	19849760.70	922:32.41	70.70
	d50r0	20091214.79	80:21.31	60.65

Table 3: Results for unrestricted path length.

Keeping this in mind, the times shown in Table 3 are reasonable, considering also the fact that the model is to be used in the annual planning process.

From Tables 3 and 4 we see that the gaps ($= 100 * \frac{upper\ bound - lower\ bound}{lower\ bound}$) are quite large. We believe that the reason for this is the weak lower bound. We have made computational experiments, too expensive to even briefly present and discuss the results here, that indicate that some of the classes of inequalities are very useful under certain parameter combinations while they lead to only minor improvements in other cases. Thus, we conclude that the big gaps can be attributed to our restricted knowledge of the facial structure of the associated polytope X.

7 Conclusions

The computational experience with our model [1] for the design of low cost survivable telecommunication networks revealed difficulties in the handling of large instances. We have presented here a modification of this model that takes into account that the capacities that can be leased in practice often have a special "divisibility property" and which results in considerably smaller problem sizes.

We are not able to solve relevant practical instances to optimality; in fact, the gaps between upper and lower bounds are still large. But, using our new model, we come up - in acceptable running times - with feasible solutions that are considered satisfactory by the network designers and that are much better than those obtained by other approaches. To achieve acceptable running times with the old model we artificially had to reduce the instance sizes in a preprocessing step. Our new model does not need such preprocessing and comes up with

		Lower bound	Time (min::sec)	Gap (%)
Network 1	d1r0	7415650.30	0:04.95	40.18
	d1r50	7408175.04	4:35.85	45.46
	d1r100	10088959.59	7:03.66	28.91
	d50r0	9397650.98	0:25.67	50.70
Network 2	d1r0	9059085.28	0:15.39	43.37
	d1r50	9089384.90	20:55.81	43.33
	d1r100	11682626.60	40:57.35	40.05
	d50r0	11256593.48	2:06.13	43.59
Network 3	d1r0	12019380.77	1:01.52	64.85
	d1r50	12261497.00	146:37.70	60.31
	d1r100	15023932.47	138:37.97	66.93
	d50r0	14562997.90	14:56.81	58.09
Network 4	d1r0	16061849.77	3:39.81	76.19
	d1r50	16191917.98	821:32.42	72.77
	d1r100	19849760.70	884:45.63	70.70
	d50r0	19619942.98	59:17.27	66.95

Table 4: Results for path length restriction $\ell_{uv} = 3$.

true upper and lower bounds on the optimum costs in case the capacities are structured in a special way. To improve the bounds we need better cutting planes, and thus, further research on the structure of the set of feasible solutions is necessary.

Acknowledgment : We would like to thank our colleagues at `E-Plus`, in particular, Dr. Peter Jonas, for fruitful discussions and excellent cooperation.

References

[1] D. Alevras, M. Grötschel, and R. Wessäly. A network dimensioning tool. Preprint SC 96-49, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 1996.

[2] F. Barahona. Network design using cut inequalities. *SIAM Journal on Optimization*, 6(3):823–837, 1996.

[3] D. Bienstock, S. Chopra, O. Günlück, and C.-Y. Tsai. Minimum cost capacity installation for multicommodity network flows. *Mathematical Programming (Special Issue on Computational Integer Programming)*, (to appear), 1995.

[4] D. Bienstock and O. Günlük. Computational experience with a difficult mixed-integer multicommodity flow problem. *Mathematical Programming*, 68:213–237, 1995.

[5] A. Bley. Node-disjoint length-restricted paths. Diplomarbeit, TU Berlin, Germany, 1997.

[6] CPLEX Optimization, Inc., Suite 279, 930 Tahoe Blvd., Bldg 802, Incline Village, NV 89451, USA. *Using the CPLEX Callable Library*, 1995. Information available via WWW at URL http://www.cplex.com.

[7] G. Dahl and M. Stoer. MULTISUN – Mathematical model and algorithms. Preprint TF R 46/92, Televerkets Forskningsinstitutt, 1992.

[8] G. Dahl and M. Stoer. A polyhedral approach to multicommodity survivable network design. *Numerische Mathematik*, 68:149–167, 1994.

[9] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to Theory of NP-completeness.* Freeman, San Francisco, 1979.

[10] M. Grötschel and C.L. Monma. Integer polyhedra associated with certain network design problems with connectivity constraints. *SIAM Journal on Discrete Mathematics*, 3:502–523, 1990.

[11] M. Grötschel, C.L. Monma, and M. Stoer. *Design of survivable networks*, volume Network Models of *Handbooks in Operations Research and Management Science*, chapter 10, pages 617–672. North-Holland, 1995.

[12] M. Iri. On an extension of the maximum-flow minimum-cut theorem to multicommodity flows. *Journal of the Operations Research Society of Japan*, 13:129–135, 1971.

[13] O. Kakusho and K. Onaga. On feasibility conditions of multicommodity flows in networks. *Transactions on Circuit Theory*, 18:425–429, 1971.

[14] A. Lisser, R. Sarkissian, and J.P. Vial. Optimal joint syntheses of base and spare telecommunication networks. Technical report, University of Genève, November 1995.

[15] T.L. Magnanti, P. Mirchandani, and R. Vachani. Modeling and solving the two–facility capacitated network loading problem. *Operations Research*, 43(1):142–156, 1995.

[16] M. Minoux. Optimum synthesis of a network with non-simultaneous multicommodity flow requirements. In P. Hansen, editor, *Studies on Graphs and Discrete Programming*, pages 269–277. North–Holland Publishing Company, 1981.

[17] Y. Pochet and L.A. Wolsey. Network design with divisible capacities: Aggregated flow and knapsack formulation. In *Proceedings IPCO2*, pages 150–164, Pittsburgh, 1992. Carnegie Mellon University.

[18] M. Stoer. *Design of Survivable Networks.* Lecture Notes in Mathematics. Springer, 1992.