

POLYHEDRAL AND COMPUTATIONAL INVESTIGATIONS FOR DESIGNING COMMUNICATION NETWORKS WITH HIGH SURVIVABILITY REQUIREMENTS

M. GRÖTSCHEL

Konrad-Zuse-Zentrum für Informationstechnik Berlin, Berlin, Germany

C. L. MONMA

Bellcore, Morristown, New Jersey

M. STOER

Telenor Research and Development, Kjeller, Norway

(Received May 1994; revision received May 1995; accepted August 1995)

We consider the important practical and theoretical problem of designing a low-cost communications network which can survive failures of certain network components. Our initial interest in this area was motivated by the need to design certain "two-connected" survivable topologies for fiber optic communication networks of interest to the regional telephone companies. In this paper, we describe some polyhedral results for network design problems with higher connectivity requirements. We also report on some preliminary computational results for a cutting plane algorithm for various real-world and random problems with high connectivity requirements, which shows promise for providing good solutions to these difficult problems.

This paper focuses on the important practical and theoretical problem of designing survivable communication networks. Our initial interest in this area was motivated by the problem of designing survivable topologies for fiber optic communication networks for the regional telephone companies; see Bellcore (1988) and Cardwell, Monma and Wu (1989) for an overview of this application and a description of a software tool developed at Bellcore. This application requires certain "two-connected" topologies so that special offices can communicate after the failure of any single network link or node.

Our earlier work on two-connected network design problems included structural properties and worst-case analysis of heuristics (Monma, Munson and Pulleybank 1990), practical heuristics (Monma and Shallcross 1989), polyhedral results (Grötschel and Monma 1990, Grötschel, Monma and Stoer 1992a), and computation with a cutting plane algorithm (Grötschel, Monma and Stoer 1992a). This naturally leads to theoretical and algorithmic questions for network design problems with higher survivability requirements. Structural properties were considered in Bienstock, Brickell and Monma (1990), and practical heuristics were considered in Ko and Monma (1989).

This paper describes polyhedral results, including natural integer programming formulations, classes of valid and facet-defining inequalities, and their associated separation problems for network design problems with higher connectivity requirements. We also report computational

results with a cutting plane algorithm on some random problems and on some real-world problems with high survivability demands. The real-world problems we obtained turned out to be quite difficult; indeed, more difficult than we expected based on our success in solving network design problems for the telephone companies. There seem to be several reasons for this:

- a. the networks are large (almost 500 nodes), sparse, and have many Steiner nodes;
- b. connectivity requirements of three or more seem to be much more difficult than two-connected problems; and
- c. the cost structure is very regular which makes proving optimality rather difficult.

In Section 1, we describe the model of network survivability that we have in mind. The polyhedral results are presented in Section 2. The cutting plane implementation and computational results are presented in Section 3.

1. MODELS OF NETWORK SURVIVABILITY

In this section, we formalize the survivable network design problem that is being considered in this paper. To do this, we need to introduce some notation.

A set V of nodes is given which represents the locations that must be interconnected into a network to provide the desired services. A collection E of edges is also specified that represents the possible pairs of nodes between which a direct link can be placed. We let $G =$

Subject classifications: Forthcoming.

Area of review: TELECOMMUNICATIONS.

(V, E) be the (undirected) graph of possible direct link connections. Each edge $e \in E$ has a nonnegative *fixed cost* c_e of establishing the direct link connection. The graph G may have parallel edges but contains no loops. The cost of establishing a network that consists of a subset $F \subseteq E$ of edges is the sum of the costs of the individual links contained in F . The goal is to build a minimum-cost network so that the required survivability conditions, which we describe next, are satisfied.

If $G = (V, E)$ is a graph, $W \subseteq V$ and $F \subseteq E$, then we denote by $G - W$ and $G - F$ the graph that is obtained from G by *deleting* the node set W and the edge set F , respectively. For notational convenience we write $G - v$ and $G - e$ instead of $G - \{v\}$ and $G - \{e\}$, respectively. The difference of two sets M and N is denoted by $M \setminus N$.

For any pair of distinct nodes $s, t \in V$, an $[s, t]$ -*path* P is a sequence of nodes and edges $(v_0, e_1, v_1, e_2, \dots, v_{l-1}, e_l, v_l)$, where each edge e_i is incident with the nodes v_{i-1} and v_i ($i = 1, \dots, l$), where $v_0 = s$ and $v_l = t$, and where no node or edge appears more than once in P . A collection P_1, P_2, \dots, P_k of $[s, t]$ -paths is called *edge-disjoint* if no edge appears in more than one path, and is called *node-disjoint* if no node (except for s and t) appears in more than one path. In particular, this implies that parallel edges between a pair of nodes s and t are considered to be node-disjoint $[s, t]$ -paths. This is in contrast to standard graph theory; but this modification of the definition was prompted by considerations that arise in practical situations. We say that a graph $G = (V, E)$ is k -*edge* (respectively, k -*node*) connected if there are k -edge (respectively, k -node) disjoint paths between every pair of distinct nodes. We note again that this definition of k -node connectivity differs from that of standard graph theory when parallel edges are present.

We measure connectivity between two nodes s and t , and between nodes lying in a given node set W . Let $G = (V, E)$ be a graph with at least two nodes and $W \subseteq V$ with $|W| \geq 2$. We set

$\lambda(G, W) :=$ the minimum cardinality of a subset F of E , such that two nodes of W are disconnected in $G - F$; and
 $\kappa(G, W) :=$ the minimum cardinality of a set $S \cup F$, where $S \subseteq V$ and $F \subseteq E$, such that two nodes of W are disconnected in $G - (S \cup F)$.

If $|W| < 2$ or if G has only one node then $\lambda(G, W)$ and $\kappa(G, W)$ are defined to be ∞ .

The survivability conditions require that the network satisfy certain edge or node connectivity requirements. To model these survivability conditions, we introduce the concept of node types. For each node $s \in V$ a non-negative integer r_s , called the *type* of s , is specified.

The network $N = (V, F)$ to be designed satisfies the *node survivability conditions* if, for each pair $s, t \in V$ of distinct nodes, N contains at least $r_{st} := \min \{r_s, r_t\}$ node disjoint $[s, t]$ -paths. Similarly, we say that $N =$

(V, F) satisfies the *edge survivability conditions* if, for each pair $s, t \in V$ of distinct nodes, N contains r_{st} edge disjoint $[s, t]$ -paths. These conditions ensure that some path between s and t will survive a prespecified level of node or link failures. We introduce additional symbols and conventions to denote these node- or edge-survivability models.

Given a graph $G = (V, E)$ and a vector of node types $r = (r_s)_{s \in V}$, we assume, without loss of generality, that there are at least two nodes of the largest type. If we say that we consider the k NCON problem (for G and r), then we mean that we are looking for a minimum-cost network that satisfies the node survivability conditions and where $k = \max \{r_s | s \in V\}$. Similarly, we speak of the k ECON problem (for G and r). Furthermore, we define

$$r(W) := \max \{r_u : u \in W\}; \text{ and}$$

$$\text{con}(W) := \min \{r(W), r(V - W)\}$$

$$= \max \{r_{st} : s \in W, t \in V - W\}$$

for any $W \subseteq V$. Let $G = (V, E)$ be a graph. For $Z \subseteq V$, let $\delta_G(Z)$ denote the set of edges with one end node in Z and the other in $V \setminus Z$. It is customary to call $\delta_G(Z)$ a *cut*. If it is clear with respect to which graph a cut $\delta_G(Z)$ is considered, we simply drop the index and write $\delta(Z)$. We also write $\delta(v)$ for $\delta(\{v\})$. If X, Y are subsets of V and $X \cap Y = \emptyset$, we set $[X : Y] := \{ij \in E | i \in X, j \in Y\}$; thus $\delta(X) = [X : V \setminus X]$. If W_1, \dots, W_p are pairwise disjoint subsets of V we denote by $[W_1 : \dots : W_p]$ the set of edges with the end nodes in different node sets. For any subset of edges $F \subseteq E$, we let $x(F)$ stand for the sum $\sum_{e \in F} x_e$. Consider the following integer linear program for a graph $G = (V, E)$ with edge costs c_e for all $e \in E$ and node types r_s for all $s \in V$:

$$\min \sum_{e \in E} c_e x_e \tag{1a}$$

subject to

$$x(\delta(W)) \geq \text{con}(W) \quad \text{for all } W \subseteq V, \emptyset \neq W \neq V; \tag{1b}$$

$$x(\delta_{G-Z}(W)) \geq \text{con}(W) - |Z| \tag{1c}$$

for all pairs $s, t \in V, s \neq t$, and

$$\text{for all } Z \subseteq V \setminus \{s, t\} \text{ with } 1 \leq |Z| \leq r_{st} - 1, \text{ and}$$

$$\text{for all } W \subseteq V \setminus Z \text{ with } s \in W, t \notin W:$$

$$0 \leq x_e \leq 1 \quad \text{for all } e \in E; \tag{1d}$$

$$x_e \text{ integral} \quad \text{for all } e \in E. \tag{1e}$$

It follows from Menger's theorem (see e.g., Chartrand and Lesniak 1986) that the feasible solutions of (1) are the incidence vectors of edge sets F such that $N = (V, F)$ satisfies all node survivability conditions; i.e., (1) is an integer programming formulation of the k NCON problem. By deleting inequalities (ii) from (1) we obtain, again from Menger's theorem, an integer programming formulation for the k ECON problem. The inequalities of type (1b) will be called *cut inequalities* and those of type (1c)

will be called *node cut inequalities*. Note that if one allows $|Z| = 0$ in the class of inequalities (1c), one would include the cut inequalities (1b).

The polyhedral approach to the solution of the k NCON (and similarly the k ECON) problem consists of studying the polyhedron obtained by taking the convex hull of the feasible solutions of (1). We set

$$kNCON(G; r) := \text{conv} \{x \in \mathbf{R}^E | x \text{ satisfies (1b, c, d, e)}\}$$

$$kECON(G; r) := \text{conv} \{x \in \mathbf{R}^E | x \text{ satisfies (1b, d, e)}\}$$

2. POLYHEDRAL RESULTS

In this section, we will take a closer look at the polyhedra $kNCON$ and $kECON$, especially at further nonredundant inequalities besides the cut and node cut inequalities. The practical aim of this theoretical investigation is to provide the basis for a cutting-plane algorithm. We assume the reader is familiar with cutting-plane methods and refer to Grötschel and Padberg (1985) and Padberg and Grötschel (1985) for an in-depth treatment. The general idea of a cutting-plane method is to solve an LP-relaxation of the problem at hand, and to generate inequalities as they are needed. The routine generating inequalities of a certain class needs to solve the following problem, for a given fractional LP-solution y .

Separation Problem (for a class C of inequalities)

Given a vector y decide whether y satisfies all inequalities in C and, if not, output an inequality violated by y .

We call an algorithm that solves an (exact) *separation algorithm* for C . The algorithm runs in polynomial time if its running time is bounded by a polynomial in $|V|$ and the encoding length of y .

We will now describe several classes of valid inequalities and indicate conditions under which some of these inequalities define facets. It turns out that results of this type are very technical.

2.1. Dimension and Trivial Inequalities

Given a graph G and node types r , an *edge e is essential with respect to $kECON(G; r)$ or $kNCON(G; r)$ if $kECON(G - e; r)$ or $kNCON(G - e; r)$, respectively, is empty. We denote by $kEES(G; r)$ the set of edges essential for $kECON(G; r)$ and by $kNES(G; r)$ the set of edges essential for $kNCON(G; r)$.*

Theorem 1. *Let $G = (V, E)$ be a graph and let $r \in \{0, 1, \dots, k\}^V$ be given.*

- Suppose that $kECON(G; r)$ is nonempty. Then $kECON(G; r) \subseteq \{x \in \mathbf{R}^E | x_e = 1 \text{ for all } e \in kEES(G; r)\}$ and $\dim(kECON(G; r)) = |E| - |kEES(G; r)|$.*
- Suppose that $kNCON(G; r)$ is nonempty. Then $kNCON(G; r) \subseteq \{x \in \mathbf{R}^E | x_e = 1 \text{ for all } e \in$*

$$kNES(G; r)\} \text{ and } \dim(kNCON(G; r)) = |E| - |kNES(G; r)|.$$

See Grötschel and Monma for the general result and proof, also of the next theorem.

Theorem 2. *Let $G = (V, E)$ be a graph and $r \in \mathbf{Z}_+^V$ such that $kECON(G; r)$ (respectively, $kNCON(G; r)$) is full-dimensional. Then*

- $x_e \leq 1$ defines a facet of $kECON(G; r)$ (respectively, $kNCON(G; r)$) for all e ;*
- $x_e \geq 0$ defines a facet of $kECON(G; r)$ (respectively, $kNCON(G; r)$) if and only if for every edge $f \neq e$ the polytope $kECON(G - \{e, f\}; r)$ (respectively, $kNCON(G - \{e, f\}; r)$) is nonempty.*

2.2. Cut Inequalities

In this section, we discuss the cut inequalities (1b) $x(\delta(W)) \geq \text{con}(W)$ for all $W \subseteq V$, $\emptyset \neq W \neq V$. The separation problem for cut inequalities can be performed in polynomial time for nonnegative vectors (we are only interested in $0 \leq y \leq 1$) by using network flow or graph connectivity algorithms as follows.

Let us consider the class \mathcal{C} of cut inequalities (1b) and let y be some vector in \mathbf{R}^E with $0 \leq y_e$ for all $e \in E$. We view the components y_e as capacities of the edges of the given graph $G = (V, E)$ and compute a Gomory-Hu tree (using the Gusfield version that consists of $|V| - 1$ calls of a max-flow algorithm and some bookkeeping; see Gomory and Hu (1961) and Gusfield (1987)). The Gomory-Hu tree has the property that, for any two nodes $u, v \in V$, the minimum capacity of a cut separating u and v is given by the smallest weight of an edge that is contained in the unique path linking u and v in the Gomory-Hu tree. Having the Gomory-Hu tree T with weights w_e for all $e \in T$ we can determine whether y satisfies all cut inequalities as follows. For each edge e , let V_e and V'_e denote the node sets of the two components of $(V, T - e)$. If there is an edge $e \in E$ with

$$w_e < \min \{ \max \{r_v | v \in V_e\}, \max \{r_v | v \in V'_e\} \}$$

then y violates the inequality $x(\delta(V_e)) \geq \text{con}(W)$, otherwise y satisfies all cut inequalities. Gomory-Hu trees can be computed quite efficiently in practice. Thus, the class of cut inequalities is very useful for the cutting plane approach from the algorithmic point of view. The class of cut inequalities also contains many facet-defining inequalities, which theoretically justifies the utilization of cut inequalities.

Hao and Orlin (1994) devised an algorithm for finding a cut of minimum capacity in a graph, which has the same running time as the max-flow algorithm. But it does not output a Gomory-Hu tree, which is needed in some of our separation routines for the more general class of partition inequalities described in subsection 2.4.

The following theorem was proven by Stoer (1992) and gives necessary and sufficient conditions for a cut inequality to define a facet for $kECON(G; r)$ when G is $(k + 1)$ -edge connected and where all nodes have the same type k .

Theorem 3. *Let $G = (V, E)$ be a $(k + 1)$ -edge connected graph, let $r_v = k$ for all nodes $v \in V$, and let $W \neq V$ be a nonempty node set. Define for each $W_i \subseteq W$ with $\emptyset \neq W_i \neq W$ the deficit of W_i as*

$$\text{def}_G(W_i) := \max \{0, k - |\delta_{G[W]}(W_i)|\}.$$

Define similarly for $U_i \subseteq V \setminus W$ with $\emptyset \neq U_i \neq V \setminus W$

$$\text{def}_G(U_i) := \max \{0, k - |\delta_{G[V \setminus W]}(U_i)|\}.$$

The cut inequality

$$x(\delta(W)) \leq k$$

defines a facet of the polytope $kECON(G; r)$ of k -edge connected graphs if and only if

- a. $G[W]$ and $G[V \setminus W]$ are connected, and
- b. for all edges $e \in E(W) \cup E(V \setminus W)$, for all pairwise disjoint node sets W_1, \dots, W_p ($p \geq 0$) of W with $W_i \neq W$ for all i , and for all pairwise disjoint node sets U_1, \dots, U_q ($q \geq 0$) of $V \setminus W$ with $U_i \neq V \setminus W$ for all i , the following inequality holds:

$$\sum_{i=1}^p \text{def}_{G-e}(W_i) + \sum_{i=1}^q \text{def}_{G-e}(U_i) - \left| \left[\bigcup_{i=1}^p W_i; \bigcup_{i=1}^q U_i \right] \right| \leq k.$$

We refer to Stoer for the details of the proof of Theorem 3 that is rather long and involved. We present a corollary that provides some simple minimal sufficient and some maximal necessary conditions on the connectivity of $G[W]$ and $G[V - W]$; these were originally proven in Grötschel and Monma.

Corollary 1. *Let $G = (V, E)$ be a $(k + 1)$ -edge connected graph, let $r_v = k$ for all nodes $v \in V$, and let $W \neq V$ be nonempty node set.*

- a. If $G[W]$ or $G[V \setminus W]$ is at most $\lfloor k/2 \rfloor$ -edge connected then $x(\delta(W)) \geq k$ does not define a facet of $kECON(G; r)$.
- b. If $G[W]$ and $G[V \setminus W]$ are k -edge connected, then $x(\delta(W)) \geq k$ defines a facet of $kECON(G; r)$.

For the case of $kECON(G; r)$ and $kNCON(G; r)$ problems with arbitrary node types, we do not know of any general results characterizing those cut inequalities that are facet-defining. However, in the case of $2ECON(G; r)$ and $2NCON(G; r)$ problems, i.e., where $r_v \in \{0, 1, 2\}$ for all nodes v , we do know necessary and sufficient conditions; see Grötschel, Monma and Stoer (1992b).

2.3. Node Cut Inequalities

In this section, we discuss the node cut inequalities (1c).

$$x(\delta_{G-Z}(W)) \geq \text{con}(W) - |Z| \tag{2}$$

for all pairs $s, t \in V, s \neq t$
 for all $Z \subseteq V \setminus \{s, t\}$ with $1 \leq |Z| \leq r_{st} - 1$, and
 for all $W \subseteq V \setminus Z$ with $s \in W, t \notin W$.

These inequalities are valid for $kNCON(G; r)$ but are not generally valid for $kECON(G; r)$.

The separation problem for the class of node cut inequalities together with the cut inequalities is solvable in polynomial time if we restrict the input to nonnegative vectors. The algorithm works as follows. Consider the class \mathcal{C} of node cut inequalities (1c) and of cut inequalities (1b), let y be some vector in \mathbb{R}^E with $0 \leq y_e$ for all $e \in E$, and let $G = (V, E)$ be the given graph. We construct a directed graph $D = (N, A)$ from G in the following way. Each node $v \in V$ is replaced by two nodes v', v'' of N that are linked by an arc (v', v'') with capacity one. Each edge $uv \in E$ is replaced by two arcs (u'', v') and (v'', u') each with capacity y_e . Let $S \subseteq V$ be a set of k nodes of maximal type, i.e., $r_s \geq r_t$ for all $s \in S$ and $t \in V \setminus S$. For all $s \in S$ and all $t \neq s$ with $r_{st} \geq 1$ we compute a minimum capacity cut C' of value c' separating s'' and t' . Such a cut is of the form $\delta^+(W') := \{(i, j) : i \in W', j \in N \setminus W'\}$, where $W' \subseteq N$ and $s'' \in W'$. The arcs in C' correspond either to arcs of the form (v', v'') for some $v \in V$ or to arcs of the type (u'', w') for some edge $uw \in E$. Without loss of generality, if $v'' \in W'$ and $v' \notin W'$ then either v'' can be moved to $N \setminus W'$ or v' can be moved to W' . We let

$$Z := \{v : (v', v'') \in C'\},$$

$$C := \{uw \in E : (u'', w') \in C'\}, \text{ and}$$

$$W := \{v \in V : v', v'' \in W'\}.$$

If $c' < r_{st}$, then we have found a node cut inequality violated by y , namely where s, t, Z , and W are as just defined and $C = \delta_{G-Z}(W)$. Possibly $Z = \emptyset$, in this case the cut inequality $x(\delta_G(W)) \geq r_{st}$ is violated. If $c' \geq r_{st}$ for all $s \in S$ and $t \neq s$ with $r_{st} \geq 1$, then there are no node cut inequalities violated by y .

We know of no general necessary and sufficient conditions for node cut inequalities to define facets of $kNCON(G; r)$, except in the low-connectivity case, i.e., where $r_v \in \{0, 1, 2\}$ for all $v \in V$; see Grötschel, Monma and Stoer (1992b). Some (complicated) necessary conditions and sufficient conditions for the general problem are derived in Stoer.

2.4. Partition Inequalities

In this section, we discuss the class of partition inequalities for $kECON$ and $kNCON$ problems that generalize the cut inequalities (1b). For a graph $G = (V, E)$ and $r \in \mathbb{Z}_+^V$, we call a collection $\{W_1, \dots, W_p\}$ ($p \geq 2$) of subsets of V a *proper partition* of V if

- $W_i \neq \emptyset$ for $i = 1, \dots, p$;
- $W_i \cap W_j = \emptyset$ for $1 \leq i < j \leq p$;
- $\cup_{i=1}^p W_i = V$;
- $r(W_i) \geq 1$ for $i = 1, \dots, p$.

Let $[W_1 : \dots : W_p]$ be the set of all edges having their endpoints in different sets W_i and let $I_1 := \{i \mid \text{con}(W_i) = 1\}$ and $I_2 := \{i \mid \text{con}(W_i) > 1\}$. Then the *partition inequality* induced by $\{W_1, \dots, W_p\}$ is defined as

$$x([W_1 : \dots : W_p]) \geq \begin{cases} p - 1, & \text{if } I_2 = \emptyset, \\ \left\lceil \frac{1}{2} \sum_{i \in I_2} \text{con}(W_i) \right\rceil + |I_1|, & \text{otherwise.} \end{cases} \tag{3}$$

It is not hard to see that the partition inequalities are valid for $k\text{ECON}(G; r)$ and $k\text{NCON}(G; r)$. Namely, if $\text{con}(W_i) \geq 2$ for $i = 1, \dots, p$, any feasible survivable subgraph has to use at least $\text{con}(W_i)$ edges of $\delta(W_i)$, so it uses at least $\sum_i \text{con}(W_i)/2$ edges of $[W_1 : \dots : W_p]$. If there are sets W_i with $\text{con}(W_i) = 1$, then a feasible solution using a minimum number α of edges of $[W_1 : \dots : W_p]$ would either use at least two edges of each $\delta(W_i)$, $i = 1, \dots, p$ (in which case, α is at least $\lceil \sum_{i \in I_2} \text{con}(W_i)/2 \rceil + |I_1|$), or $|\delta(W_i)| = 1$ for some W_i (in which case induction on $|I_1|$ may be used by deleting W_i from the graph G and the feasible solution). So partition inequalities are valid. The separation problem for partition inequalities is known to be NP-hard; see Grötschel, Monma and Stoer (1992a).

We know of no general necessary and sufficient conditions for partition inequalities to define facets of $k\text{ECON}(G; r)$ or $k\text{NCON}(G; r)$. Some special cases are dealt with in Grötschel and Monma (1990) and Stoer. We state here one particularly nice result of Stoer.

Theorem 4. *Let $G = (V, E)$ be a complete graph with $k + 1$ parallel edges between each pair of distinct nodes and let $r \in \mathbb{Z}_+^V$. The partition inequality (3) given by a proper partition W_1, W_2, \dots, W_p of V with $p \geq 2$ defines a facet of $k\text{ECON}(G; r)$ if and only if at least one of the following conditions hold:*

- a. $\sum_{i \in I_2} \text{con}(W_i)$ is odd;
- b. $I_1 \neq \emptyset$; or
- c. $p = 2$.

One tool in the proof of Theorem 4 was an algorithm for constructing feasible solutions for $k\text{ECON}(G; r)$ with a minimum number of edges. This is interesting in its own right. The construction procedure is a modification of an algorithm by Chou and Frank (1970).

Algorithm

Given an instance (G, r) of the $k\text{ECON}$ problem, where

- $G = (V, E)$ is a complete graph of p nodes with at least $k + 1$ parallel edges between each pair of nodes,

- $r_v \geq 2$ for all $v \in V$, $\sum_{v \in V} r_v$ is even, and there are two nodes of the highest type k ,

find a subgraph N of G , feasible for the $k\text{ECON}$ problem, using a minimum number of edges of G , namely $\sum_{v \in V} r_v/2$.

STEP 1. Choose any order v_1, \dots, v_p of the nodes in V which is imagined to be cyclic. If the highest node type k is even and there are nodes of odd type, the order should be chosen in such a way that there are two nodes (say u and v) of odd type that separate two nodes (say x and y) of type k in the cycle formed by the edges $v_1v_2, v_2v_3, \dots, v_pv_1$; i.e., in traversing the cycle, we encounter these four nodes on the order u, x, v, y . (Necessarily there are at least two nodes of highest type k and at least two nodes of odd type, if any.)

STEP 2. Construct a subgraph N' from the empty set by adding a cycle through all nodes of type at least i respecting the given order for $i = 2, 4, \dots, k$ if k is even (or $i = 2, 4, \dots, k - 1$ if k is odd).

STEP 3. Let $\{w_1, w_2, \dots, w_l\}$ be the set of all nodes of odd type numbered in the way they are met when scanning through $\{v_1, v_2, \dots, v_p\}$ in the given order. (The starting node of the scan does not matter, nor does the direction. Note also that the number l of nodes of odd type must be even.) Construct the desired subgraph N by adding to N' the matching consisting of edges $w_iw_{i/2+i}$ ($i = 1, \dots, l/2$).

It can be shown that the separation problem for partition inequalities is NP-hard, even if we restrict attention to vectors y with $0 \leq y \leq 1$; see Grötschel, Monma and Stoer (1992a). However, there are fast and successful heuristics for the separation of partition inequalities; and our computational experiments have revealed that partition inequalities are very helpful for solving network survivability problems; see Section 3.

2.5. Node Partition Inequalities

In this section, we consider the class of node partition inequalities which generalize the node cut inequalities in a manner similar to how the partition inequalities (3) generalize the cut inequalities (1b). Consider a graph $G = (V, E)$ and $r \in \mathbb{Z}_+^V$. Let $Z_2 \subseteq \dots \subseteq Z_k \subseteq V$, $k \geq 2$ be node sets with $|Z_j| = j - 1$ for $j = 2, \dots, k$. Let $\{W_1, \dots, W_p\}$ be a proper partition of $V \setminus Z_k$, such that at least two node sets in the partition contain nodes of type k . Define $I_j := \{i \mid r(W_i) \geq j\}$ for $j = 1, \dots, k$. The node partition inequality induced by W_1, \dots, W_p and Z_2, \dots, Z_k is given by

$$x([W_1 : \dots : W_p : Z_k]) - x(\cup_{j=2}^k \cup_{i \in I_j} [Z_j : W_i]) \geq p - 1. \tag{4}$$

Theorem 5. *The node partition inequalities (4) are valid for $k\text{NCON}(G; r)$.*

Proof. In the case where $p = 2$, the node partition inequality (4) defines a node cut inequality which is valid (see Section 1). The case $p \geq 3$ is treated by induction.

So let $\{W_1, \dots, W_p\}$ and Z_2, \dots, Z_k induce a node partition inequality with $p \geq 3$. Let W_p be the node set with the smallest value of $r(W_i)$. Define $j := r(W_p)$. Then for $i = 1, \dots, p - 1$ the node partition inequality induced by Z_2, \dots, Z_k and $\{W_1, W_2, \dots, W_i \cup W_p, \dots, W_{p-1}\}$ is valid and has a right-hand side of $p - 2$. Note that none of the coefficients of the left-hand side of this new inequality is larger than in the original inequality, and that the coefficients of edges in $[W_i : W_p]$ have dropped to 0. Adding all these node partition inequalities and the node cut constraint $x(\delta_{G-Z_j}(W_p)) \geq 1$ gives an inequality $a^T x \geq (p - 1)(p - 2) + 1$, where all coefficients of a^T have a value of at most $p - 1$. Dividing this inequality by $p - 1$ and rounding up the right-hand side and all coefficients of the left-hand side produces the desired node partition inequality with the right-hand side $p - 1$.

It is known to be NP-hard to separate this class of inequalities; see Grötschel and Monma. Some necessary and some sufficient conditions for these inequalities to define facets are known only in a few special cases; see Grötschel, Monma and Stoer (1992b) and Stoer (1992).

It can be shown that the separation problem for partition inequalities is NP-hard, even if we restrict attention to vectors y with $0 \leq y \leq 1$; see Grötschel, Monma and Stoer (1992a). However, there are fast and successful heuristics for the separation of partition inequalities; and our computational experiments have revealed that partition inequalities are very helpful for solving network survivability problems; see Section 3.

2.6. r -Cover and Lifted r -Cover Inequalities

A nice combinatorial relaxation of the k ECON (and then the k NCON problem) is the r -cover problem that can be defined as follows. Given a graph $G = (V, E)$ and positive integers r_v for all $v \in V$, an r -cover is a set $F \subseteq E$ of edges such that $|F \cap \delta(v)| \geq r_v$ for all $v \in V$. Clearly, every solution of an ECON problem defined by a graph G and node types $r \in N^V$ is an r -cover. Hence, if edge weights are given in addition, a lower bound for the ECON problem can be determined by solving the r -cover problem which means finding a r -cover of minimum weight. This relaxation is of particular interest since the r -cover problem can be solved in polynomial time. The polynomial-time solvability follows from the fact that the r -cover problem can be transformed into a (1-capacitated) b -matching problem.

In his paper, Edmonds (1965) not only gave a polynomial time algorithm for solving the b -matching problems, he also determined a complete linear description of the b -matching polytope. Edmonds' blossom inequalities for the 1-capacitated b -matching polytope of a graph $G = (V, E)$ can be transformed to the r -cover case. This

transformation (see Grötschel, Monma and Stoer 1991), yields the r -cover inequalities, valid for $\text{ECON}(G; r)$, that have the form:

$$x(E(H)) + x(\delta(H)\setminus T) \geq \left\lceil \left(\sum_{v \in H} r_v - |T| \right) / 2 \right\rceil \tag{5}$$

for all $H \subseteq V$, and all $T \subseteq \delta(H)$.

The separation problem for the class of r -cover inequalities and vectors in the unit hypercube can be solved in polynomial time as follows. Suppose that $x^* \in Q^E, x^* \geq 0$ is given. We define a new vector $y^* := 1 - x^*$ and a 1-capacitated b -matching problem by setting $b_v := \deg(v) - r_v$ for all $v \in V$. Then it is easy to see that x^* satisfies all r -cover inequalities (5) if and only if y^* satisfies the blossom inequalities for the corresponding b -matching problem which read:

$$y(E(H)) + y(T) \leq \left\lfloor \left(\sum_{v \in H} b_v + |T| \right) / 2 \right\rfloor \tag{6}$$

for all $H \subseteq V$ and all $T \subseteq \delta(H)$.

Padberg and Rao (1982) have designed a polynomial time algorithm for the separation problem for the class of blossom inequalities (6). This algorithm is based on the Gomory-Hu procedure for finding a minimum cut in a graph with nonnegative edge capacities. If the Padberg-Rao algorithm yields an inequality

$$y(E(H)) + y(\bar{T}) \leq \left\lfloor \left(\sum_{v \in H} b_v + |T| \right) / 2 \right\rfloor$$

that is violated by y^* then x^* violates the r -cover inequality

$$x(E(H)) + x(\delta(H)\setminus T) \geq \left\lceil \left(\sum_{v \in H} r_v - |T| \right) / 2 \right\rceil.$$

In case some of the nodes have type 1 the r -cover inequalities can be strengthened as:

$$x(E(H)) + x(\delta(H)\setminus T) \geq \left\lceil \left(\sum_{v \in H, r_v \geq 2} r_v - |T| \right) / 2 \right\rceil + |\{v \in H | r_v = 1\}|. \tag{7}$$

These inequalities are valid for k ECON($G; r$) but not for the r -cover polytope. To solve the separation problem for the class of strengthened r -cover inequalities (7) heuristically we do the following. We declare all nodes of type 1 to be of type 2 and apply the procedure described before for the modified node types. If the Padberg-Rao separation algorithm finds a violated blossom inequality, then the corresponding strengthened r -cover inequality (7) is violated. However, this trick does not yield an exact separation routine because there may be violated inequalities of type (7) that this procedure does not find. This is due to the fact that the transformation of node types may violate one of the requirements for the Padberg-Rao procedure to work correctly; i.e., this method is a heuristic for strengthened r -cover inequalities (7) in case nodes of type 1 are present.

We now generalize the r -cover inequalities further. Let $G = (V, E)$ be a graph and $r \in \{0, \dots, k\}^V$. Let $H \neq V$ be a node set, called the *handle*, and let $T \subseteq \delta(H)$. For each $e \in T$ we denote by T_e the set of the two end nodes of e . The sets $T_e, e \in T$, are called *teeth*. For simplicity we also call the edges $e \in T$ teeth in this section. If an edge $e \in T$ is parallel to some edge $f \in T$, we count T_e and T_f as two sets, even if $T_e = T_f$. Let $\{H_1, \dots, H_p\}, p \geq 3$, be a partition of H into nonempty disjoint node sets such that

- $r(H_i) \geq 1$ for $i = 1, \dots, p$;
- no more than $\text{con}(H_i) - 1$ teeth intersect any $H_i, i = 1, \dots, p$;
- at least three H_i are intersected by teeth;
- $\sum_{i \in I_2} \text{con}(H_i) - |T|$ is odd, where $I_2 := \{i \mid \text{con}(H_i) \geq 2\}$.

Let $I_1 := \{i \mid \text{con}(H_i) = 1\}$. We call

$$x([H_1 : \dots : H_p]) + x(\delta(H) \setminus T) \geq \frac{1}{2} \left(\sum_{i \in I_2} \text{con}(H_i) - |T| + 1 \right) + |I_1| \tag{8}$$

the *lifted r -cover inequality* induced by H_1, \dots, H_p, T .

It is not hard to show (see Stoer) that the lifted r -cover inequalities (8) are valid for $k\text{ECON}(G; r)$ and hence for $k\text{NCON}(G; r)$. Some necessary conditions are known for these inequalities to define facets for $k\text{ECON}(G; r)$, and one can easily see that these inequalities do not define facets for $k\text{NCON}(G; r)$ for “highly” connected graphs G ; see Stoer. It is known that the separation problem for lifted r -cover inequalities is NP-hard; see Grötschel, Monma and Stoer (1992a). We have produced a separation heuristic for class (8) that uses some graph manipulation techniques (such as the shrinking of edges) and is based on an analysis of the Gomory–Hu tree, as provided by the Padberg–Rao algorithm (1982).

2.7. Further Remarks on Valid Inequalities

We note that we know more general versions of some of the classes of valid inequalities introduced before, and we also know additional classes of inequalities valid for $\text{ECON}(G; r)$ or $\text{NCON}(G; r)$. We have omitted the introduction of these classes here because many technical definitions are necessary to describe them, and we do not have general results about the dimensions of the faces that they define, and we could not make any algorithmic use of them so far. Some information about these classes can be found in Stoer.

There is a rich literature in polyhedral combinatorics on facet manipulation techniques, i.e., methods by which new types of facets can be derived from known classes. These techniques usually run under the name “facet lifting” or “facet extension.” A number of lifting results are shown in Stoer. Included are techniques such as: addition of an edge, addition of a node, and expansion of a node w into a node set W where all edges in $E(W)$

receive a coefficient zero in the lifted inequality. The description of these techniques and the proofs of the associated lifting results are rather technical and not presented here.

Our results about characterizing those inequalities among the classes of valid inequalities described before that are facet-defining for $k\text{ECON}(G; r)$ or $k\text{NCON}(G; r)$ appear somewhat unsatisfactory. In fact, they are. But having worked on the facial structure of these polytopes for some time we are convinced that general results covering large classes of graphs and node types simultaneously are very hard to obtain. Theorem 3 gives a glimpse at the technical subtleties involved in a seemingly simple-looking case. The main difficulty we see is that a slight change of the graph or node type may result in a considerable change of the problem complexity and the polyhedral structure. For instance, if $r_v = 1$ for all $v \in V$, $k\text{ECON}(G; r)$ is equal to $k\text{NCON}(G; r)$, which is nothing but the convex hull of the incidence vectors of all the supersets of spanning trees of G ; changing a few node types to zero we obtain an NP-hard Steiner tree problem; setting $r_v = 2$ for all $v \in V$, $2\text{NCON}(G; r)$ and $2\text{ECON}(G; r)$ contain the traveling salesman polytope on G .

Based on our observations, we think that more investigation should go into the study of more restricted, practically relevant cases and not into the investigation of the whole range of $k\text{ECON}$ or $k\text{NCON}$ polytopes.

3. A CUTTING PLANE ALGORITHM AND COMPUTATIONAL RESULTS

In this section, we give an outline of our cutting plane algorithm for the $k\text{ECON}$ and $k\text{NCON}$ problems. We describe it for the $k\text{NCON}$ problem. The algorithm for the $k\text{ECON}$ problem is derived from the $k\text{NCON}$ problem by skipping all those separation routines that check inequalities that are valid for $k\text{NCON}(G; r)$ but not for $k\text{ECON}(G; r)$.

Our cutting plane procedure starts with solving the LP:

$$\begin{aligned} &\text{minimize } c^T x && (9) \\ &\text{subject to } x(\delta(v)) \geq r_v && \text{for all } v \in V \text{ with } r_v \geq 1; \\ & && 0 \leq x_e \leq 1 \text{ for all } e \in E \end{aligned}$$

consisting of at most $|V|$ degree inequalities and the $2|E|$ trivial inequalities. Almost all of these define facets of $k\text{NCON}(G; r)$, if $k\text{NCON}(G; r)$ is full-dimensional (see Theorem 1). An optimal solution $y \in \mathbf{R}^E$ of this relaxation of the $k\text{NCON}$ problem is usually not feasible for the polytope $k\text{NCON}(G; r)$. (If it were, we would be finished.)

So, in each iteration of the cutting plane algorithm we try to find inequalities (more specifically: cut, node cut, partition, node partition, and lifted r -cover inequalities) that are valid for $k\text{NCON}(G; r)$, but are violated by y . Geometrically, such an inequality defines a hyperplane in

R^E separating y from the k ECON-polyhedron, a so-called "cutting plane." The heuristics and exact algorithms for finding inequalities violated by a given y are called *separation routines*.

We add all the violated inequalities found by our separation routines to the current LP and solve the revised LP to get a new optimum solution y . (We do not solve the new LP from scratch, but use postoptimization.) We repeat this process until the current optimal LP solution y happens to be feasible for k NCON($G; r$), or no further inequalities violated by y are found. In the second case, we proceed with a branch-and-cut method.

In the first case (y feasible), we know that y is optimal, because the present LP is a relaxation of the k NCON problem. Note that feasibility of y is identical with y being a $\{0, 1\}$ -vector that satisfied all cut constraints and node-cut constraints. This feasibility criterion is easy to check.

Since we are using only a subset of all facet-defining inequalities for k NCON($G; r$), we cannot be sure to find an optimal solution with such a cutting plane algorithm for all graphs G , cost functions c , and node types r . In any case, even if the present fractional solution y is not feasible, its objective function value $c^T y$ provides a lower bound for the k NCON problem, which is increased with every iteration (or at least, it does not drop).

We summarize the cutting plane algorithm as follows.

Algorithm (Cutting Plane Algorithm for k NCON)

1. Solve the LP (9). Let y be an optimal solution.
2. While y is not feasible for k NCON($G; k$) do:
 - a. find valid inequalities violated by y , add them to the LP, delete some redundant inequalities, and resolve the LP. Let y be a new optimal solution.
 - b. If no violated inequalities can be found, perform branch and cut.

In 2a we first determine violated lifted r -cover inequalities, when no violated partition inequalities can be found. At present, we have a preliminary version of a code for solving survivability problems with higher connectivity requirements. To test our code for general k NCON problems, we first used a set of random problems. Later, we also obtained test data for a real-world 3NCON problem, which arose in the design of a communication network on a ship. Both types of test problems have their "drawbacks," however. Most random problems turned out to be too easy, and the ship problem confronted us with many new difficulties.

We first report about our computational results on random problems. We used the same set of random data as Ko and Monma used for their high-connectivity heuristics. So we will be able to compare results later. All running times reported are on a SUN 4/50 IPX workstation. The LP-solver used is a research version of the CPLEX-code provided to us by Bixby (1992). This is a very fast implementation of the simplex algorithm.

The test set consists of five complete graphs of 40 nodes and five complete graphs of 20 nodes, whose edge costs are independently drawn from a uniform distribution of real numbers between 0 and 20. For each of these ten graphs, a minimum-cost, k -edge connected subgraph for $k = 3, 4, 5$ is to be found. The next table reports the number of iterations (minimum and maximum) and the average time (in seconds) taken by our code to solve these problems for $k = 3, 4$, and 5, respectively. Only the time for the cutting plane phase is given.

# Nodes $K =$	# Iterations			Average Time (Secs.)		
	3	4	5	3	4	5
20	1-2	1-5	1-4	0.43	0.51	0.58
40	1-2	1-2	1-4	1.54	1.95	2.36

All problems except one 3ECON instance on 20 nodes were solved in the cutting plane phase. In fact, 20 of the 30 problems were already solved in the first iteration with the initial LP (9). For the instances not solved in the first iteration, at most four lifted r -cover inequalities (8) had to be added to obtain the optimal solution. Except for one 3ECON instance, no partition inequalities were added. So, the average solution time is mainly the solution time for the first LP.

All optimal solutions for the k ECON problems were at the same time feasible and hence optimal for the corresponding k NCON problems, except the one 3ECON problem which could not be solved in the cutting plane phase. There the optimal solution (obtained by branch and cut) is 3-edge connected, but not 3-node connected.

These excellent results were surprising, because we always thought high-connectivity problems to be harder than low-connectivity problems. But this does not seem to be true for random costs.

The high-connectivity heuristics of Ko and Monma performed reasonably well. The relative gap between the heuristic (h) and the optimal solution value (o), namely $100 \times (h - o)/o$, computed for the previous set of random problems, ranged between 0.8 and 12.8 with an average of 6.5% error (taken over all problems).

The second set of test problems were five complete graphs whose 40 nodes were placed randomly in a square, and whose edge weights are the Euclidean distances between the end nodes. All nodes are of the same type k , where k ranges between 1 and 5, so five k ECON problems are derived from each of the five graphs. The 1ECON problems were solved with a spanning tree algorithm.

Table I shows the computational results for these problems. Its entries are:

- K the required connectivity;
- P the number of partition inequalities used in the cutting plane phase;
- RC the number of lifted r -cover inequalities used in the cutting plane phase;

Table I
Performance of Branch and Cut on Euclidean Problems

K	IT	P	RC	COPT	Gap	TT
1				77.20-95.31		
2	3-10	54-70	0- 8	90.69-107.72	0.00-0.23	0-1
3	3- 6	55-74	4-24	156.83-184.95	2.12-2.65	33-324
4	1- 2	40-46	0- 4	221.64-264.39	0.00	0-1
5	1- 8	40-44	0-15	307.12-363.19	0.00-0.98	0-186

IT the number of iterations in the cutting plane phase;

COPT the optimal value;

GAP the relative error between the optimal value and the lower bound achieved by the cutting plane phase;

TT the total running time (with branch and cut, etc.) in minutes.

All k -edge connectivity problems with even k , except one 2ECON problem, could be solved in the cutting plane phase. The running time for the cutting plane phase was at most six seconds for all problems. Of the 5ECON problems, two could be solved in the cutting plane phase in two iterations. Two other 5ECON problems were solved by branch and cut in at most three minutes, and only one took 186 minutes. For all 3ECON problems, branch and cut had to be employed to bridge the gap. The relatively large gaps in the 3ECON problems indicate that our separation routines should be enhanced to find the partition and r -cover inequalities still violated by the last LP-solution and that, maybe, other facet-defining classes of inequalities for the k ECON polytope for odd k are needed. Because of this poor performance and because of the nonavailability of good upper bounds, the branching procedure degenerated into enumeration for several instances.

Considering the gaps and the running times, we may conclude that k -edge connectivity problems for even k are easier to solve than those for odd $k \geq 3$, and that the k -edge connectivity problems with odd k become easier for "large" k . One reason for the difference between even and odd k may be that, for even k , the only facet-defining partition inequalities are the cut inequalities, which are easy to find.

Concerning the structure of the solutions, all optimal solutions for even k were regular graphs, except for one 2ECON instance, where two nodes of degree 3 appeared. For $k \geq 3$, the solutions are not necessarily k -node connected. Some are only 2-node connected.

The optimal values can be said to be roughly linearly increasing in k , from $k \geq 2$. By the way, the lower values in column COPT of Table I are all due to the same problem instance, and the upper values too.

To give an impression of the solution structures, Figure 1 was included, depicting, for one of the graphs and $k = 1, \dots, 4$, the optimal k ECON solutions in clockwise order.

One real-world application of survivable network design, where connectivities higher than two are needed, is the design of a fiber communication network that connects locations on a military ship containing various communication systems. The reason for demanding high survivability of this network is obvious.

The problem of finding a high-connected network topology minimizing the cable installation cost can be formulated as a 3NCON problem. We will describe the characteristics of this problem in what follows.

We obtained the graph and edge cost data of a generic ship model. It has the following features. The graph of possible link installations has the form of a three-dimensional grid with 15 layers, 494 nodes, and 1,096 edges, which is depicted in Figure 2.

The ship problem comes with three different options: the node types may vary; the costs may be normal or random; and the underlying graph may be reduced or not. Of the grid's 494 nodes, there are 461 regular nodes (depicted by hollow circles), 30 special nodes (depicted by filled circles) in the main part of the ship, and three

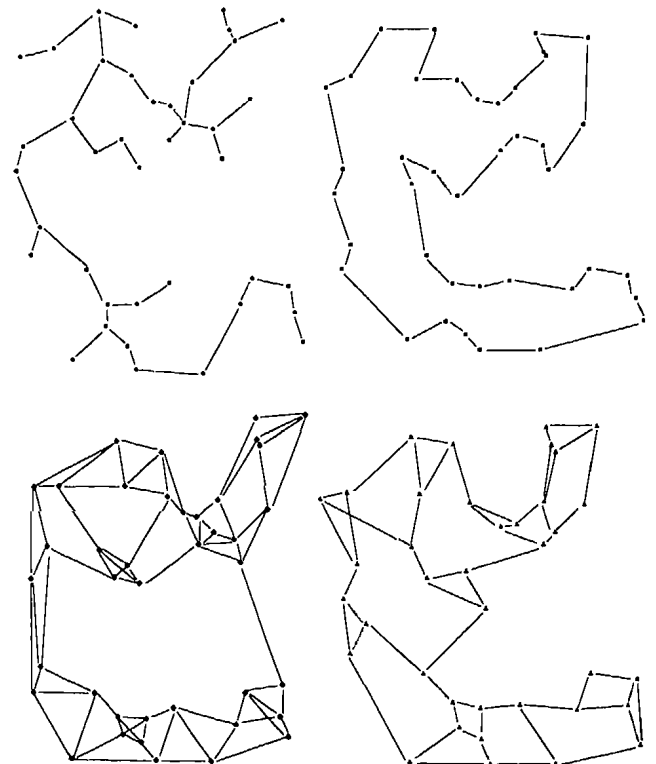


Figure 1. Solution of Euclidean problems for $k = 1-4$.

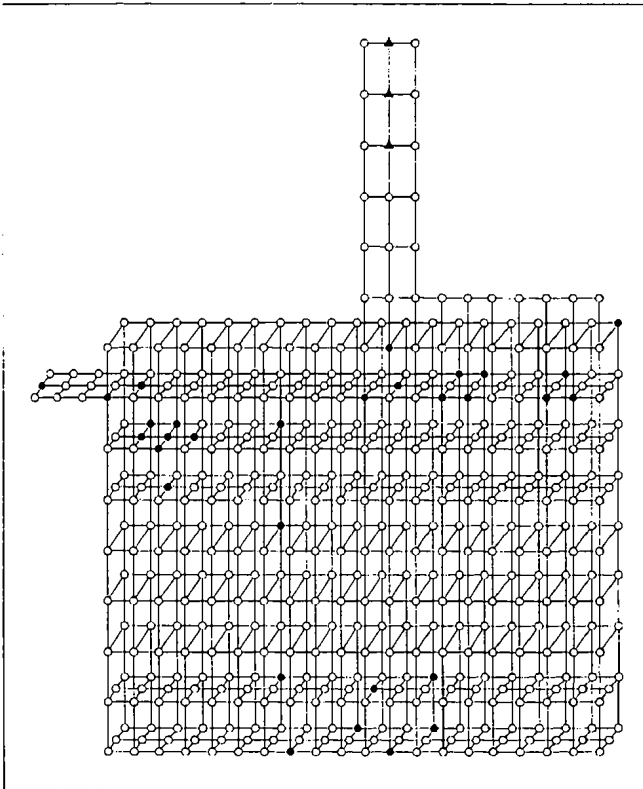


Figure 2. Grid graph of the ship problem.

priority nodes (depicted by triangles) in the ship's tower. The special and priority nodes represent various communication, command and control systems to be interconnected by the network. The regular nodes represent fiber junction boxes where the fiber cable can be routed. The notation "shipxyz" will be used to indicate that the regular nodes are of type x , special nodes are of type y and priority nodes are of type z . So ship013 is the problem, where the three nodes in the tower are of type 3, the 30 special nodes in the body of the ship are of type 1, and all other 461 grid nodes are of type 0.

No branch and cut was used because of the long computation times for some of the problems. Inequalities that are not almost tight are removed from the LP, but are still kept in a pool of inequalities. But since even this pool grew too much, inequalities are removed from there that have not been violated for more than five iterations in a row. All problems were solved as ECON problems. No node partition inequalities were used.

The upper bounds were produced by a heuristic for NCON problems (not ECON) that tries to eliminate as many edges from the input graph as possible, starting with the first edge. (This heuristic was written by Bill Cook.) The input graph was the graph of edges formed by the support of the best fractional solution, and the edges were ordered by increasing value of the fractional solution. (Not by increasing costs, as all horizontal edges have the same cost.)

The normal cost structure is highly regular. The costs are roughly proportional to the distances between nodes,

with the feature that horizontal distances are much longer than vertical distances. (The grid shown in Figure 2 has been scaled. Also, contrary to the graphical representation, the horizontal layers do not always have the same distance from each other.) With this cost structure, it is much cheaper to route vertically than horizontally. Since there exist many shortest paths between any two nodes, there will also exist many optimum solutions to the survivable network problem. So the problem is highly degenerate. Degeneracy together with the size of the ship problem caused us to run into difficulties. We also considered "random" uniform $[0, 1]$ costs which were scaled so that the overall cost of the edges remained the same as in the original problem. These problems were indicated by appending "rand" to the problem name.

We considered "reduced" versions of these problems where we removed some of the "unnecessary" nodes in the lower left and right hand corner of the grid, and also deleted some of the horizontal layers of the grid containing only nodes of type 0. It is not obvious at all that corners of a grid may be cut out and layers may be deleted without affecting the optimum objective function value of the problem. But nevertheless, we used these reductions heuristically to cut down problem sizes in the hope that some optimal solution of the original graph is still contained in the reduced graph. For the ship023 problem, this hope was confirmed. These reduced problems were denoted by appending red to the problem name.

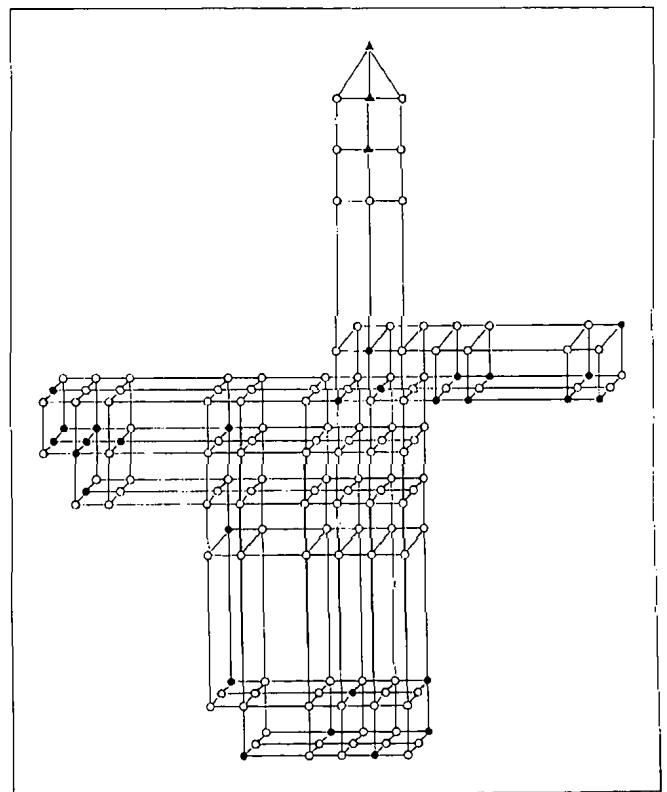


Figure 3. Reduced grid graph of the ship13 problem.

Table II
SIZES OF SHIP PROBLEMS

Problem	Original Graph					Reduced Graph				
	0	1	2	3	Edges	0	1	2	3	Edges
ship013	461	30	0	3	1096/0	128	28	0	3	325/6
ship023	461	0	30	3	1096/0	249	0	30	3	607/6
ship033	461	0	0	33	1096/0	300	0	0	33	719/9

Figure 3 shows the reduced graph of the **ship013** problem. The result of the reductions can be seen from Table II, whose columns list, from left to right, the problem names, and, for the original ship graph and the reduced ship graphs, the number of nodes of type 0, 1, 2, and 3, the total number of nodes and the total number of edges/number of forced edges. The forced edges are those edges contained in some cut of size 3 separating two nodes of type 3, which must be contained in any feasible solution. Table II shows that the reductions are enormous, yet there are still many more nodes of type 0 than nodes of nonzero type in each problem.

When we applied our code to the reduced graphs, the fractional solutions still looked frequently like paths beginning at some special node and ending in some node of type 0. To cure this problem, we made use of the following type of inequalities

$$x(\delta(v) \setminus \{e\}) \geq x_e$$

for all nodes v of type 0 and all $e \in \delta(v)$.

These inequalities (we call them *con0 inequalities*) describe algebraically that nodes of type 0 do not have degree 1 in an edge-minimal solution. This is not true for all survivable networks, but it is true for the optimum solution if all costs are positive. So, although these inequalities are not valid for the k NCON polytope, we

used them to force the fractional solutions into the creation of longer paths.

Table III gives some computational results of our cutting plane algorithm on the several versions of the ship problem. The entries from left to right are:

- Problem Problem name with "red" for reduced and "rand" for random costs;
- VAR The number of edges minus the number of forced edges;
- IT The number of iterations (i.e., LPs solved);
- Part The number of partition inequalities (3) added;
- RCOV The number of lifted r -cover inequalities (8) added;
- LB The lower bound (i.e., optimal LP value);
- Gap (UB-LB)/LB in percent;
- Time In minutes : seconds.

Table IV gives a breakdown of how time was spent. The entries from left to right are:

- Problem Problem name with "red" for reduced and "rand" for random costs;
- PT Time spent for reduction of the problem (in percent);
- LPT Time spent for LP solving (in percent);
- CT Time spent for separation (in percent);

Table III
Performance of the Cutting Plane Algorithm on the Ship Problems

Problem	VAR	IT	Part	RCOV	LB	UB	Gap	Time
ship013	1,088	3,252	777,261	0	211957.1	217428	2.58	10122:35
ship013red	322	775	200,570	0	217428	217428	0	426:47
ship013rand	1,088	1,273	301,190	0	171409.8	171409.9	0	2783:15
ship023	1,088	15	4,090	0	286274	286274	0	27:20
ship023red	604	12	2,372	0	286274	286274	0	1:54
ship023rand	1,088	11	3,649	0	245905.8	248249.0	0.94	42:31
ship033	1,082	42	10,718	1	461590.6	483052	4.64	55:26
ship033red	710	40	9,817	0	462099.3	483052	4.53	34:52
ship033rand	1,082	89	23,505	0	422169.6	428535.5	1.50	198:17
ship113	1,090	128	17,199	0	902691.0	918691	1.77	4724:55
ship113rand	1,090	45	4,343	0	789280.7	817352.1	3.55	13:33
ship123	1,088	61	13,210	0	906691.0	930691	2.57	1167:37
ship123rand	1,088	1,942	54,846	9	834951.9	856666.7	2.60	694:20
ship133	1,084	176	21,564	0	945052.0	1008808	6.74	119:15
ship133rand	1,084	582	22,788	0	941568.2	969578.7	2.97	195:11
ship223	1,085	5	541	0	940925.0	940925	0	0:44
ship223rand	1,085	5	625	0	1090417.5	1090430.0	0.001	0:54
ship233	1,081	5	532	0	1028193.0	1029176	0.09	0:54
ship233rand	1,081	10	599	6	1172230.3	1183636.9	0.97	1:53

Table IV
Computation Time on the Ship Problems

Problem	PT	LPT	CT	MT	Time
ship013	0.0	75.6	23.9	0.5	10122:35
ship013red	0.0	68.5	30.1	1.4	426:47
ship013rand	0.0	64.0	35.1	0.9	2783:15
ship023	0.0	13.1	86.4	0.4	27:20
ship023red	0.1	39.2	58.6	1.9	1:54
ship023rand	0.0	2.7	97.1	0.2	42:31
ship033	0.0	31.2	68.2	0.6	55:26
ship033red	0.0	41.1	58.4	0.5	34:52
ship033rand	0.0	27.6	71.9	0.4	198:17
ship113	0.0	98.4	1.6	0.0	4724:55
ship113rand	0.0	42.0	56.8	1.2	13:33
ship123	0.0	97.9	2.1	0.0	1167:36
ship123rand	0.0	23.9	75.7	0.5	694:19
ship133	0.0	60.4	38.9	0.7	119:15
ship133rand	0.0	32.0	67.3	0.7	195:11
ship223	0.3	18.7	76.7	4.5	0:44
ship223rand	0.2	15.9	79.8	4.1	0:54
ship233	0.2	22.4	74.5	2.9	0:54
ship233rand	0.1	9.5	88.7	1.7	1:53

MT Time spent on miscellaneous items, input, output, etc. (in percent);

Time Total time in minutes : seconds.

Several problems were solved to optimality: **ship013** (normal costs, reduced), **ship023** (normal costs, reduced and not reduced), and **ship223** (normal costs). The **ship223** problem with normal costs ended up with a fractional solution of value 940,925, but the primal heuristic found an integer solution with the same value.

Reduction paid off by a factor of at least 20 in the computing times of the **ship013** and **ship023** problems. The reduction of the **ship023** problem did not affect the optimal value.

The largest gaps are for the **ship033** and **ship133** problems for normal costs, reduced or not reduced. More polyhedral investigation is probably required in these cases. By inspection, we found a few more violated partition inequalities. So, also better separation routines for partition inequalities are needed.

The problems **ship013** (not reduced, normal and random costs) and **ship113** (not reduced, normal costs) took longest. The solution of **ship013** (not reduced, normal costs) is distributed over the whole net with many small fractionals. This shows that tree-like problems are not well handled. Problems with many nodes of type 2 are easier than problems without nodes of type 2.

Lifted r -cover inequalities (very few, however) were found only for the **ship033**, **ship123**, **ship233** problems. Therefore, better separation routines for r -cover inequalities for this special kind of graph are needed.

As one can see from Table III our code is still painfully slow for the problems involving many nodes of type 1. In each iteration, only small progress is made. Although many inequalities are added, they do not produce big structural changes in the fractional solution. A better

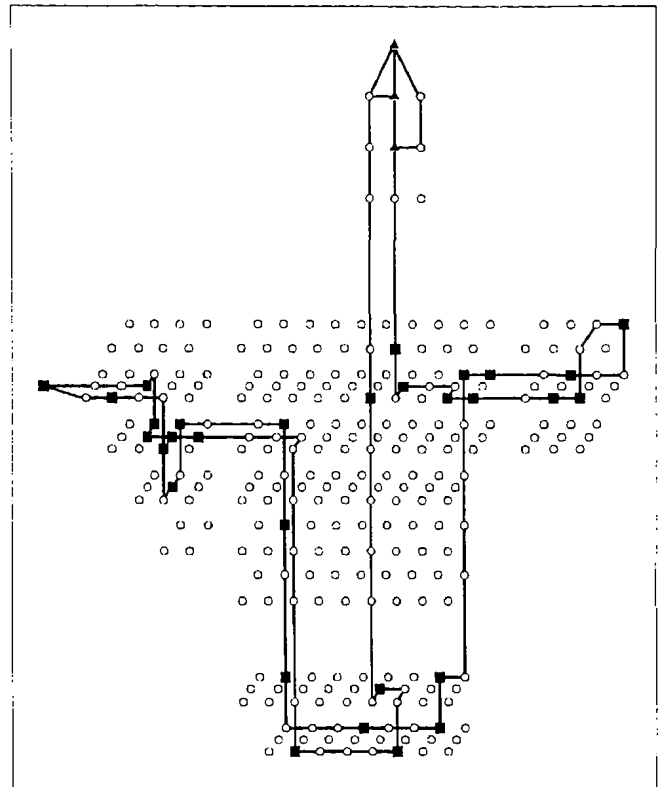


Figure 4. Optimum solution of the reduced **ship23** problem.

strategy might be to add structured sets of partition or cut inequalities that somehow anticipate the “escape maneuvers” of the fractional solution.

An optimal solution to the reduced **ship023** problem is shown in Figure 4. We do not understand yet why our code solves the **ship023** problem rather easily and why there is still a gap after a long running time of our cutting plane algorithms for other problems. Probably, the “small” changes of a few survivability requirements result in more dramatic structural changes of the polyhedra and thus of the inequalities that should be used. It is conceivable that our code has to be tuned according to different survivability requirements settings.

To summarize our computational results, for survivability problems with high connectivity requirements, many nodes of type 0 and highly regular cost structure (such as the ship problems) much still remains to be done to speed up our code and enhance the quality of solutions. This is in contrast to our previous work (see Grötschel, Monma and Stoer 1992a) on applications in the area of telephone network design, where problem instances typically are of moderate size and contain not too many nodes of type 0, and where our approach produces very good lower bounds and even optimum solutions in a few minutes. Yet, we see our work as a promising step toward solving problems with high-connectivity constraints.

REFERENCES

- BELLCORE. 1988. FIBER OPTIONS, Software for designing survivable optimal fiber networks. Software Package, December 1988.
- BIENSTOCK, D., E. F. BRICKELL AND C. L. MONMA. 1990. On the Structure of Minimum-Weight k -Connected Spanning Networks. *SIAM J. Discr. Math.* **3**, 320-329.
- BIXBY, R. E. 1992. Implementing the Simplex Method: The Initial Basis. Tech. *ORSA J. Comput.* **4**, 267-284.
- CARDWELL, R. H., C. L. MONMA AND T-H. WU. 1989. Computer-Aided Design Procedures for Survivable Fiber Optic Networks. *IEEE J. Selected Areas Commun.* **7**, 1188-1197.
- CHARTRAND, C., AND L. LESNIAK. 1986. *Graphs & Digraphs* (2nd ed.). Wadsworth & Brooks/Cole, Pacific Grove, Calif.
- CHOU, W., AND M. FRANK. 1970. Survivable Communication Networks and the Terminal Capacity Matrix. *IEEE Trans. Circ. Theory* **CT-17**, 183-192.
- EDMONDS, J. 1965. Maximum Matching and a Polyhedron With 0, 1 Vertices. *J. Res. Natl. Bur. Stand.* **69**, 125-130.
- GOMORY, R. E., AND T. C. HU. 1961. Multi-terminal Network Flows. *SIAM J. Appl. Math.* **9**, 551-570.
- GRÖTSCHEL, M., AND C. L. MONMA. 1990. Integer Polyhedra Associated With Certain Network Design Problems With Connectivity Constraints. *SIAM J. Discr. Math.* **3**, 502-523.
- GRÖTSCHEL, M., C. L. MONMA AND M. STOER. 1991. Polyhedral Approaches to Network Survivability. In *Reliability of Computer and Communication Networks*, F. Roberts, F. Hwang, and C. L. Monma (eds.). AMS/ACM, Series in Discrete Mathematics and Computer Science, **5**, 121-141.
- GRÖTSCHEL, M., C. L. MONMA AND M. STOER. 1992. Computational Results With a Cutting Plane Algorithm for Designing Communication Networks With Low-Connectivity Constraints. *Opns. Res.* **40**, 309-330.
- GRÖTSCHEL, M., C. L. MONMA AND M. STOER. 1992. Facets for Polyhedra Arising in the Design of Communication Networks With Low-Connectivity Constraints. *SIAM J. Optim.* **2**, 474-504.
- GRÖTSCHEL, M., AND M. W. PADBERG. 1985. Polyhedral Theory. Chapter 8 In *The Traveling Salesman Problem* E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, D. B. Shmoys (eds.). Wiley, Chichester, U.K., 251-305.
- GUSFIELD, D. 1990. Very Simple Methods for all Pairs Network Flow Analysis. *SIAM J. Comput.* **19**, 143-155.
- HAO, J., AND J. B. ORLIN. 1994. A Faster Algorithm for Finding the Minimum Cut in a Directed Graph. *J. Algor.* **17**, 424-446.
- KO, C. W., AND C. L. MONMA. 1989. Heuristic Methods for Designing Highly Survivable Communication Networks, Technical Report, Bellcore, Morristown, N.J.
- MONMA, C. L., B. S. MUNSON AND W. R. PULLEYBLANK. 1990. Minimum-Weight Two-Connected Spanning Networks. *Math. Prog.* **46**, 153-171.
- MONMA, C. L., AND D. F. SHALLCROSS. 1989. Methods for Designing Communication Networks With Certain Two-Connected Survivability Constraints. *Opns. Res.* **37**, 531-541.
- PADBERG, M. W., AND M. GRÖTSCHEL. 1985. Polyhedral Computations. Chapter 9, In *The Traveling Salesman Problem*, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys (eds.). Wiley, Chichester, U. K. 307-360.
- PADBERG, M., AND R. RAO. 1982. Odd Minimum Cut Sets and b -Matchings. *Math. O.R.* **7**, 67-80.
- STOER, M. 1992. Design of Survivable Networks. Ph.D. Thesis, University of Augsburg. (Appeared in *Lecture Notes in Mathematics*, **1531**. Springer, Heidelberg, Germany, 1992.)